

FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER, L.L.P.
1300 I STREET, N. W.
WASHINGTON, DC 20005-3315

202 • 408 • 4000
FACSIMILE 202 • 408 • 4400

WRITER'S DIRECT DIAL NUMBER:

(202) 408-4024

September 22, 2000

ATLANTA
404 • 653 • 6400
PALO ALTO
650 • 849 • 6600

TOKYO
011 • 813 • 3431 • 6943
BRUSSELS
011 • 322 • 646 • 0353

#2
Jc714 U.S. PTO
09/667769
09/22/00

ATTORNEY DOCKET NO.: 04329.2431

Box Patent Application
Assistant Commissioner for Patents
Washington, D.C. 20231

New U.S. Patent Application

Title: METHOD FOR DETERMINING A SERVER COMPUTER WHICH
CARRIED OUT A PROCESS MOST RECENTLY, AND HIGH AVAILABILITY
COMPUTER SYSTEM

Inventors: Kotaro ENDO and Koji YAMAMOTO

Sir:

We enclose the following papers for filing in the United States Patent and
Trademark Office in connection with the above patent application.

1. A check for \$730.00 representing the filing fee and \$40.00 for recording the Assignment.
2. Application - 64 pages, including 3 independent claims and 15 claims total.
3. Drawings - 5 sheets of formal drawings containing 7 figures.
4. Declaration and Power of Attorney.
5. Recordation Form Cover Sheet and Assignment to Kabushiki Kaisha Toshiba.
6. Certified copy of Japanese Patent Application No. 11-364571, filed on December 22, 1999.

Best Available Copy

FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER, L.L.P.

Assistant Commissioner for Patents
September 22, 2000
Page 2

Applicants claim the right to priority based on Japanese Patent Application No. 11-364571, filed on December 22, 1999.

Please accord this application a serial number and filing date and record and return the Assignment to the undersigned.

The Commissioner is hereby authorized to charge any additional filing fees due and any other fees due under 37 C.F.R. § 1.16 or § 1.17 during the pendency of this application to our Deposit Account No. 06-0916.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

By: 

Richard V. Burgujian
Reg. No. 31,744

RVB/FPD/mld
Enclosures

日 本 国 特 許 庁

PATENT OFFICE
JAPANESE GOVERNMENT

Jc714 U.S. PTO
09/667769
09/22/00

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日

Date of Application:

1 9 9 9 年 1 2 月 2 2 日

出 願 番 号

Application Number:

平成 1 1 年 特 許 願 第 3 6 4 5 7 1 号

出 願 人

Applicant (s):

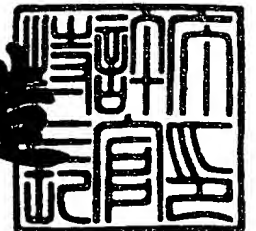
株式会社東芝

CERTIFIED COPY OF
PRIORITY DOCUMENT

2 0 0 0 年 8 月 4 日

特 許 庁 長 官
Commissioner,
Patent Office

及 川 耕 造



出 証 番 号 出 証 特 2 0 0 0 - 3 0 6 2 2 2 3

【書類名】 特許願

【整理番号】 A009906608

【提出日】 平成11年12月22日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 15/16
G06F 17/00
H04L 29/14

【発明の名称】 最後に処理を行っていたサーバ計算機を判定するプログラムを記録した記録媒体、及び高可用性計算機システム

【請求項の数】 8

【発明者】

【住所又は居所】 東京都府中市東芝町 1 番地 株式会社東芝府中工場内

【氏名】 山本 浩司

【発明者】

【住所又は居所】 東京都府中市東芝町 1 番地 株式会社東芝府中工場内

【氏名】 遠藤 浩太郎

【特許出願人】

【識別番号】 000003078

【氏名又は名称】 株式会社 東芝

【代理人】

【識別番号】 100058479

【弁理士】

【氏名又は名称】 鈴江 武彦

【電話番号】 03-3502-3181

【選任した代理人】

【識別番号】 100084618

【弁理士】

【氏名又は名称】 村松 貞男

【選任した代理人】

【識別番号】 100068814

【弁理士】

【氏名又は名称】 坪井 淳

【選任した代理人】

【識別番号】 100092196

【弁理士】

【氏名又は名称】 橋本 良郎

【選任した代理人】

【識別番号】 100091351

【弁理士】

【氏名又は名称】 河野 哲

【選任した代理人】

【識別番号】 100088683

【弁理士】

【氏名又は名称】 中村 誠

【選任した代理人】

【識別番号】 100070437

【弁理士】

【氏名又は名称】 河井 将次

【手数料の表示】

【予納台帳番号】 011567

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 最後に処理を行っていたサーバ計算機を判定するプログラムを記録した記録媒体、及び高可用性計算機システム

【特許請求の範囲】

【請求項 1】 2 台のサーバ計算機的一方が処理を行い、処理を行っているサーバ計算機で障害が発生した場合に、もう一方のサーバ計算機が処理を引き継ぐことが可能な高可用性計算機システムに適用され、前記各サーバ計算機に実行させるための最後に処理を行っていたサーバ計算機を判定するプログラムを記録した計算機読み取り可能な記録媒体であって、

前記各サーバ計算機に、

前記 2 台のサーバ計算機の少なくとも一方で障害が発生した場合と障害から復帰した場合とに、所定の状態遷移図に従って当該サーバ計算機の状態遷移を行う状態遷移ステップと、

前記状態遷移に応じて、少なくとも当該遷移後の自サーバ計算機の状態で決まるサーバ優先度を自サーバ計算機がローカルに持つ記憶装置に記録するサーバ優先度記録ステップと、

前記 2 台のサーバ計算機の両方に障害が発生し、その後少なくとも自サーバ計算機が障害から復帰した際に、少なくとも当該自サーバ計算機の前記記憶装置に記録されたサーバ優先度に基づいて当該自サーバ計算機の方が優先度が高いか否かを判定する優先度判定ステップと、

前記優先度判定ステップでの優先度判定結果に基づいて、自サーバ計算機が最後に処理を行っていたサーバ計算機であるか否かを判定することで、自サーバ計算機が処理を継続するか否かを決定し、その決定結果により前記状態遷移ステップでの状態遷移を行わせる最終処理計算機判定ステップと

を実行させるための最後に処理を行っていたサーバ計算機を判定するプログラムを記録した記録媒体。

【請求項 2】 前記優先度判定ステップは、前記 2 台のサーバ計算機の両方に障害が発生し、その後両サーバ計算機が共に障害から復帰した際に、当該両サーバ計算機の前記記憶装置に記録されたサーバ優先度を比較して、自サーバ計算

機の方が優先度が高いか否かを判定する第 1 の優先度判定ステップを含むことを特徴とする請求項 1 記載の最後に処理を行っていたサーバ計算機を判定するプログラムを記録した記録媒体。

【請求項 3】 前記優先度判定ステップは、前記 2 台のサーバ計算機の両方に障害が発生し、その後自サーバ計算機のみが障害から復帰した際に、自サーバ計算機の前記記憶装置に記録されたサーバ優先度が最高優先度であるか否かを判定する最高優先度判定ステップと、前記最高優先度判定ステップで最高優先度であると判定された場合にのみ自サーバ計算機の方が優先度が高いと判定する第 2 の優先度判定ステップを更に含むことを特徴とする請求項 2 記載の最後に処理を行っていたサーバ計算機を判定するプログラムを記録した記録媒体。

【請求項 4】 前記状態遷移ステップは、前記各サーバ計算機の状態を、処理を行い、且つ行っている処理を引き継ぐ相手が存在する「マスタ」と、処理は行いが、行っている処理を引き継ぐ相手が存在しない「シングルマスタ」と、処理を行っていないが、引き継ぎのための情報を受け取っている「スレーブ」と、処理を行っておらず、且つ引き継ぎのための情報を受け取っていない「停止」の 4 つの状態に分類して作成された前記状態遷移図に従って前記 2 台のサーバ計算機の状態遷移を行うことを特徴とする請求項 3 記載の最後に処理を行っていたサーバ計算機を判定するプログラムを記録した記録媒体。

【請求項 5】 前記状態遷移ステップは、前記 2 台のサーバ計算機の両方に障害が発生し、その後両サーバ計算機が共に障害から復帰した際に、前記最終処理計算機判定ステップでの判定結果と前記状態遷移図とに基づいて、前記 2 台のサーバ計算機的一方が前記停止状態から前記シングルマスタ状態となり、他方が前記停止状態から前記スレーブ状態となる状態遷移を行う第 1 の状態遷移ステップと、その後前記他方がスレーブ状態のまま、前記一方がマスタ状態となる状態遷移を行う第 2 の状態遷移ステップとを含むことを特徴とする請求項 4 記載の最後に処理を行っていたサーバ計算機を判定するプログラムを記録した記録媒体。

【請求項 6】 前記状態遷移ステップは、前記 2 台のサーバ計算機の両方に障害が発生し、その後自サーバ計算機のみが障害から復帰した際に、前記最終処理計算機判定ステップでの判定結果と前記状態遷移図とに基づいて、自サーバ計

算機が前記停止状態から前記シングルマスタ状態となる状態遷移を行うか、或いは現在の状態を維持する第 3 の状態遷移ステップと、その後相手サーバ計算機が障害から復帰した際に、当該相手サーバ計算機が前記停止状態から前記スレーブ状態となる状態遷移を行うか、或いは前記 2 台のサーバ計算機的一方が前記シングルマスタ状態となり、他方が前記スレーブ状態となる状態遷移を行う第 4 の状態遷移ステップとを含むことを特徴とする請求項 4 記載の最後に処理を行っていたサーバ計算機を判定するプログラムを記録した記録媒体。

【請求項 7】 前記状態遷移ステップは、前記 2 台のサーバ計算機の両方に障害が発生し、その後自サーバ計算機のみが障害から復帰した際に、当該自サーバ計算機の前記記憶装置に記録されたサーバ優先度が最高優先度でないために、前記最高優先度判定ステップでの判定ができず、当該自サーバ計算機が処理を継続する計算機であると判定できない状態で、当該自サーバ計算機に対して強制的に処理を継続させる強制開始命令が外部から与えられた場合に、当該自サーバ計算機が処理を継続するサーバ計算機として当該自サーバ計算機の状態遷移を行わせる強制開始ステップを含むことを特徴とする請求項 3 記載の最後に処理を行っていたサーバ計算機を判定するプログラムを記録した記録媒体。

【請求項 8】 2 台のサーバ計算機的一方が処理を行い、処理を行っているサーバ計算機で障害が発生した場合に、もう一方のサーバ計算機が処理を引き継ぐことが可能な高可用性計算機システムにおいて、

前記各サーバ計算機は、

前記 2 台のサーバ計算機の少なくとも一方で障害が発生した場合と障害から復帰した場合とに、所定の状態遷移図に従って当該サーバ計算機の状態遷移を行う状態遷移手段と、

前記状態遷移手段により状態遷移が行われる度に、少なくとも当該遷移後の自サーバ計算機の状態で決まるサーバ優先度を自サーバ計算機がローカルに持つ記憶装置に記録する状態書き込み手段と、

前記 2 台のサーバ計算機の両方に障害が発生し、その後少なくとも自サーバ計算機が障害から復帰した際に、少なくとも当該自サーバ計算機の前記記憶装置に記録されたサーバ優先度に基づいて当該自サーバ計算機の方が優先度が高いか否

かを判定する優先度判定手段と、

前記優先度判定手段での優先度判定結果に基づいて、自サーバ計算機が最後に処理を行っていたサーバ計算機であるか否かを判定することで、自サーバ計算機が処理を継続するか否かを決定し、その決定結果により前記状態遷移手段による状態遷移を行わせる最終処理計算機判定手段と

を具備することを特徴とする計算機システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、2台のサーバ計算機的一方が処理を行い、処理を行っているサーバ計算機で障害が発生しても、もう一方のサーバ計算機が処理を引き継ぐことができる高可用性計算機システムに係り、特に両方のサーバ計算機に障害が発生し、その後少なくとも一方が障害から復帰した場合に、その復帰した計算機が処理を継続するか否かを判定するのに必要な、最後に処理を行っていたサーバ計算機を判定するプログラムを記録した記録媒体、及び高可用性計算機システムに関する

。

【0002】

【従来の技術】

従来から、複数のサーバ計算機（以下、サーバと略称する）、例えば2台のサーバをネットワーク等で結合し、1つのサーバで障害が発生しても、障害で停止した処理（サービス）を別のサーバが引き継ぐことにより、システム全体として可用性を維持できるようにしたクラスタ型の耐障害コンピュータシステム、つまり高可用性計算機システムが種々開発されている。

【0003】

この種の計算機システムのうち、共有ディスク装置等の共有記憶装置を備えたシステムでは、一方のサーバで処理を実行しているときに、当該サーバから処理を引き継ぐのに必要な情報を当該共有記憶装置に記録しておくのが一般的である

。

【0004】

このような構成の計算機システムでは、例えば2台のサーバに障害が発生して、その後両サーバが共に障害から復帰した場合、或いは、いずれか一方のサーバだけが障害から復帰した場合のいずれの場合にも、共有記憶装置に記録された情報を利用することで、どの障害復帰サーバであっても容易に引き継ぎを行うことができる。

【0005】

ところが、高可用性計算機システムの中には、共有記憶装置を持たないシステムもある。このような計算機システムでは、処理の引き継ぎを可能とするために、一方のサーバで処理を実行しているときに、当該サーバから他方のサーバに対し、処理の続きをするのに必要な情報を送るのが一般的である。

【0006】

このようにすると、処理を実行しているサーバに障害が発生して処理を続行できなくなった場合に、他方のサーバでそれまでに送られてきた情報を使って処理の続きを行うこと、つまり処理を引き継ぐことが可能となる。

【0007】

しかし、両方のサーバに障害が発生した場合に上記の処理の引き継ぎを行うのは容易ではない。その理由は、例えば両サーバが共に障害復帰（復旧）した場合であれば、いずれのサーバが処理を行う（引き継ぐ）かを決定しなければならないからである。また、いずれか一方のサーバだけが障害から復帰した場合であれば、その復帰したサーバで処理を行う（引き継ぐ）かを判定しなければならないからである。この従来技術について、以下に詳述する。

【0008】

2台のサーバに障害が発生して、両サーバが共に障害復帰した場合、それまでに処理を行っていなかったサーバ（スレーブ）が処理の続きを行うためには、処理を行っていたサーバ（マスタ）から引き継ぎのための情報をもらっておく必要がある。

【0009】

ところが、最後に処理を行っていたサーバ（マスタ）に障害が発生する直前に、このサーバからもう一方のサーバ（スレーブ）に対して、引き継ぎのための情

報を送っていなかった場合は、当該もう一方のサーバ（スレーブ）では処理の続きを行うことができない。そのため、両サーバが共に障害復帰した際には、最後に処理を行っていた方のサーバを、処理を引き継ぐサーバとして決定（選択）する必要がある。

【 0 0 1 0 】

次に、最後に処理を行っていたサーバ（マスタ）に障害が発生する前に、このサーバからもう一方のサーバ（スレーブ）に対して、引き継ぎのための情報を送っていた場合は、両サーバが共に障害復帰した際に、どちらのサーバを選択しても処理の続きをすることが可能である。しかし、障害発生直前の処理については、最後に処理を行っていたサーバから引き継ぎに必要な情報をもう一台のサーバに送る前に障害が発生している可能性がある。そのため、こちらの場合も、最後に処理を行っていた方のサーバを選択する必要がある。

【 0 0 1 1 】

また、いずれか一方のサーバだけが障害から復帰した場合にも、そのサーバに処理をさせる条件は、両サーバが共に障害復帰した場合と同様の理由により、そのサーバが2台のうち最後まで処理を行っていたサーバであることである。

【 0 0 1 2 】

そのため従来は、最後に処理を行っていたサーバが判定可能なように、次に述べる2つの方法のいずれかを適用していた。

【 0 0 1 3 】

(1) 引き継ぎを1度に限定する方法

予め、どちらか1つのサーバをプライマリサーバ、もう1つのサーバをセカンダリサーバと決めておき、初めは必ずプライマリサーバがマスタ、セカンダリサーバがスレーブとして動作を開始する。ここで、マスタは処理（クライアントから要求された処理）を行い、引き継ぎのための情報をスレーブに送る。スレーブはマスタから送られる引き継ぎのための情報を受け取って自身の有するディスク装置等の外部記憶装置（ローカルな外部記憶装置）に保存する。

【 0 0 1 4 】

そして、1度、プライマリサーバに障害が発生して引き継ぎを行ったなら、そ

の後はプライマリサーバが復帰しても利用しない。つまり、引き継ぎを1度に限定する。この場合、セカンダリサーバが、自身の外部記憶装置に自身が処理を行ったかどうかを記録しておくことにすれば、容易にどちらが最近まで処理を行っていたかを判定できる。

【0015】

しかし、この従来技術は引き継ぎが1度しかできないため、片方または両方のサーバに障害発生、或いは障害復帰がいつ起こっても、可能な限り処理を継続するという、いわゆる自動運転が実現できない。

【0016】

(2) 時刻情報を使用する方法

予め、2台のサーバの時計(時刻))を合わせておく。サーバが処理を開始するときに、現在時刻を自身の持つ外部記憶装置に記録する。このようにすると、両サーバが共に障害復帰した際に、外部記憶装置に記録した時刻の情報をネットワークを通じて互いに授受し、より新しい時刻情報を持つサーバを、最後に処理を行っていたサーバとして判定する。

【0017】

この時刻情報を使う方法は、時間というグローバルなものを暗黙に仮定しており、各サーバの使う時計が常に同期しているということを仮定している。しかし、実際の時計は必ずしも常に同期しているとは限らず、この方法は完全性に問題がある。また、一方のサーバだけが障害復帰した場合には、もう一方のサーバとの間で時刻情報の授受が行えないため、自身が最後に処理を行っていたサーバであるか否か判定できない。

【0018】

【発明が解決しようとする課題】

上記したように、2台のサーバの一方が処理を行い、処理を行っているサーバで障害が発生しても、もう一方のサーバが処理を引き継ぐことが可能な高可用性計算機システムであって、共有記憶装置を持たない従来の高可用性計算機システムでは、2台のサーバに障害が発生し、その後少なくとも一方が障害から復帰した場合に、処理を継続するサーバ、即ち最後に処理を行っていたサーバを判定可

能とする方法として、「引き継ぎを 1 度に限定する方法」、或いは「時刻情報を使用する方法」が適用されていた。

【0019】

しかし、「引き継ぎを 1 度に限定する方法」では、引き継ぎが 1 度しかできないため、自動運転が実現できないという問題があった。

また、「時刻情報を使用する方法」では、時間というグローバルなものを暗黙に仮定し、各サーバの使う時計が常に同期しているということを仮定しているが、実際の時計は必ずしも常に同期しているとは限らず、したがって判定精度（完全性）に問題があった。また、一方のサーバだけが障害復帰した場合には、自身が最後に処理を行っていたサーバであるか否か判定できないという問題もあった。

【0020】

本発明は上記事情を考慮してなされたものでその目的は、時刻情報といったグローバルな情報を使わず、ローカルな情報だけを使って最後に処理を行っていたサーバを判定することにより、処理引き継ぎの回数の制限をなくして自動運転を実現すると共に、判定精度の向上が図れる、最後に処理を行っていたサーバ（サーバ計算機）を判定するプログラムを記録した記録媒体、及び高可用性計算機システムを提供することにある。

【0021】

【課題を解決するための手段】

本発明の記録媒体は、2 台のサーバの一方が処理を行い、処理を行っているサーバで障害が発生した場合に、もう一方のサーバが処理を引き継ぐことが可能な高可用性計算機システムに適用される、最後に処理を行っていたサーバを判定するプログラムであって、上記各サーバに、以下の各ステップ、即ち、上記 2 台のサーバの少なくとも一方で障害が発生した場合と障害から復帰した場合とに、所定の状態遷移図に従って当該サーバの状態遷移を行う状態遷移ステップと、この状態遷移に応じて、少なくとも当該遷移後の自サーバの状態で決まるサーバ優先度を自サーバがローカルに持つ記憶装置に記録するサーバ優先度記録ステップと、サーバの両方に障害が発生し、その後少なくとも自サーバが障害から復帰した

際に、少なくとも当該自サーバの記憶装置に記録されたサーバ優先度に基づいて当該自サーバの方が優先度が高いか否かを判定する優先度判定ステップと、この優先度判定ステップでの優先度判定結果に基づいて、自サーバが最後に処理を行っていたサーバであるか否かを判定することで、自サーバが処理を継続するか否かを決定し、その決定結果により上記状態遷移ステップでの状態遷移を行わせる最終処理計算機判定ステップとを実行させるための、最後に処理を行っていたサーバを判定するプログラムを記録したことを特徴とする。

【0022】

このような構成においては、所定の状態遷移図に従う状態の遷移で決まるサーバ優先度という状態変数がローカルな記憶装置に記録され、当該状態遷移とローカルな状態変数を用いて、最後まで処理を行っていたサーバが判定されることから、処理引き継ぎの回数の制限をなくして自動運転を実現することが可能となる。

【0023】

ここで、上記2台のサーバの両方に障害が発生し、その後両サーバが共に障害から復帰した際の処理のために、上記優先度判定ステップに次のステップ、即ち、上記両サーバが共に障害から復帰した際に、当該両サーバのサーバ優先度を比較して、自サーバの方が優先度が高いか否かを判定する第1の優先度判定ステップを持たせるとよい。

【0024】

また、上記2台のサーバの両方に障害が発生し、その後自サーバのみが障害から復帰した際の処理のために、上記優先度判定ステップに次の2つのステップ、即ち、自サーバのサーバ優先度が最高優先度であるか否かを判定する最高優先度判定ステップと、ここで最高優先度であると判定された場合にのみ自サーバの方が優先度が高いと判定する第2の優先度判定ステップとを持たせるとよい。

【0025】

また本発明は、上記状態遷移図により示される上記2台のサーバの状態として、処理を行い、且つ行っている処理を引き継ぐ相手が存在する「マスタ」と、処理は行いが、行っている処理を引き継ぐ相手が存在しない「シングルマスタ」と

、処理を行っていないが、引き継ぎのための情報を受け取っている「スレーブ」と、処理を行っておらず、且つ引き継ぎのための情報を受け取っていない「停止」の4つの状態の組み合わせを適用したことを特徴とする。

【0026】

このように、処理を行うサーバを、行っている処理を引き継ぐ相手が存在する「マスタ」と、行っている処理を引き継ぐ相手が存在しない「シングルマスタ」とに分類して区別することで、両サーバのローカルな状態変数（サーバ優先度）が同一値となるのを防止して、当該ローカルな状態変数を用いて最後に処理を行っていたサーバを判定する場合の判定精度を向上することが可能となる。

【0027】

この効果は、上記サーバ優先度記録ステップに次の4つのステップ、即ち、自サーバの状態がシングルマスタ状態に遷移する場合に、自サーバのサーバ優先度を最高優先度を示すように変更する第1のサーバ優先度記録ステップと、自サーバの状態がマスタ状態に遷移する場合に、自サーバのサーバ優先度を2番目の優先度を示すように変更する第2のサーバ優先度記録ステップと、自サーバの状態がスレーブ状態に遷移する場合に、自サーバのサーバ優先度を最低優先度を示すように変更する第3のサーバ優先度記録ステップと、自サーバの状態が停止状態に遷移する場合に、自サーバのサーバ優先度の変更を抑止する（元優先度を再度記録することと等価）サーバ優先度維持ステップとを持たせることにより、一層顕著となる。

【0028】

上記4つのサーバ状態を適用した構成では、2台のサーバの両方に障害が発生し、その後両サーバが共に障害から復帰した際に、上記最終処理計算機判定ステップでの判定結果と状態遷移図とに基づいて、上記2台のサーバの一方が停止状態からシングルマスタ状態となり、他方が停止状態からスレーブ状態となる状態遷移を行い、その後上記他方がスレーブ状態のまま、上記一方がマスタ状態となる状態遷移を行うとよい。

【0029】

また、2台のサーバの両方に障害が発生し、その後自サーバのみが障害から復

帰した際には、上記最終処理計算機判定ステップでの判定結果と状態遷移図とに基づいて、自サーバが停止状態からシングルマスタ状態となる状態遷移を行うか、或いは現在の状態を維持し、その後相手サーバが障害から復帰した際に、当該相手サーバが停止状態からスレーブ状態となる状態遷移を行うか、或いは一方がシングルマスタ状態となり、他方がスレーブ状態となる状態遷移を行うとよい。

【 0 0 3 0 】

また本発明は、上記状態遷移ステップに次のステップ、即ち、上記 2 台のサーバの両方に障害が発生し、その後自サーバのみが障害から復帰した際に、当該自サーバのサーバ優先度が最高優先度でないために、上記最高優先度判定ステップでの判定ができず、当該自サーバが処理を継続する計算機であると判定できない状態で、当該自サーバに対して強制的に処理を継続させる強制開始命令が外部から与えられた場合に、当該自サーバが処理を継続するサーバとして当該自サーバの状態遷移を行わせる強制開始ステップを持たせたことをも特徴とする。

【 0 0 3 1 】

このような構成においては、2 台のサーバに障害が発生し、その後 1 台だけが障害復帰しても、その 1 台のサーバ優先度が最高優先度でないために、その 1 台の方が停止状態にあるもう 1 台より優先度が高いと判定することが不可能で、したがって、その 1 台が最後に処理を行っていたサーバであると判定できない場合でも、その 1 台に対して外部の計算機（コマンド送信用計算機）から強制開始命令を与えることで、強制的に処理を行わせることが可能となる。このように強制的に処理を開始させることができる機能は、処理の続きは無理でも、処理を再開した方が都合がよい場合に便利である。

【 0 0 3 2 】

ここで、停止状態にあるサーバの優先度が最高優先度のときに、障害復帰したサーバに対して強制的に処理を行わせると、両サーバが共に最高優先度となる。このような状態では、両サーバに共に障害が発生して、その後共に障害復帰した場合に、優先度の高いサーバを特定できず、したがって処理の引き継ぎを自動的に行うことはできないが、再度強制開始を行うことで容易に対処可能となる。

【 0 0 3 3 】

なお、上記した最後に処理を行っていたサーバを判定するプログラムを記録した記録媒体に係る本発明は、各計算機が当該プログラムで実現される機能手段を備えた高可用性計算機システムに係る発明としても成立し、当該プログラムで適用される手順を持つ方法に係る発明としても成立する。

【 0 0 3 4 】

【発明の実施の形態】

以下、本発明の実施の形態につき図面を参照して説明する。

図 1 は本発明の一実施形態に係る高可用性計算機システムの構成を示すブロック図である。

【 0 0 3 5 】

図 1 のシステムにおいて、ネットワーク 5 0 0 には、2 台のサーバ（サーバ計算機）1 0 0 a, 1 0 0 b と、コマンド送信用計算機 3 0 0 とが接続されている。またネットワーク 5 0 0 には、サーバ 1 0 0 a または 1 0 0 b からのサービスを受ける図示せぬクライアント（クライアント計算機）も接続されている。サーバ 1 0 0 a, 1 0 0 b には、それぞれ外部記憶装置としての（ハードディスク装置に代表される）ディスク装置 2 0 0 a, 2 0 0 b が接続されている。この計算機システムは共有記憶装置を持たない。

【 0 0 3 6 】

図 1 の計算機システムは、2 台のサーバ 1 0 0 a, 1 0 0 b のうち的一方が処理を行い、処理を行っているサーバで障害が発生しても、もう一方のサーバが処理を引き継ぐことができるだけでなく、両サーバ 1 0 0 a, 1 0 0 b で共に障害が発生して、その後少なくとも一方が障害から復帰した場合にも、処理を引き継ぐサーバ、つまり最後に処理を行っていたサーバを正しく判定することが可能な仕組みを有していることに特徴がある。この仕組みについては後述する。

【 0 0 3 7 】

なお、本実施形態における「障害の発生」とは、ハードウェア障害、ソフトウェア障害などの他に、電源の切断などを含めて、処理の続行ができなくなる状態をいう。また、「障害からの復帰」とは、ハードウェア障害からの復帰、ソフトウェア障害からの復帰の他に、電源の投入などを含めて、処理の続行が行える状

態になることをいう。

【0038】

さて、処理の引き継ぎを可能とするためには、単に同じ処理を別のサーバで行うだけではなく、それまで処理を行っていたサーバから、処理の続きを行うのに必要な情報を受け取っておかなければならない。ところが本実施形態のように共有記憶装置を持たないシステムでは、当該共有記憶装置を介してのサーバ間の情報授受ができないことから、処理を行うサーバが、引き継ぎに必要な情報を他方のサーバに送らなければならない。しかし、この方式では、障害の発生時期と引き継ぎに必要な情報の転送時期との関係で、[従来の技術]の欄で述べたように、最後に処理を行っていたサーバを判定することが困難となる場合がある。そこで、最後に処理を行っていたサーバが容易に判定できる方法として、[従来の技術]で述べた「引き継ぎを1度に限定する方法」、或いは「時刻情報を使用する方法」が知られているが、自動運転が実現できないとか判定精度（完全性）が不十分であるといった問題がある。

【0039】

ここで自動運転について詳述する。

自動運転とは、[従来の技術]の欄で述べたように、片方または両方のサーバに障害発生、或いは障害復帰がいつ起こっても、可能な限り処理を継続する運転方法を指す。自動運転を実現するためには、少なくとも次の動作（1）～（5）が行える必要がある。

【0040】

（1）2台のサーバに障害が発生し、その後2台のサーバが復帰したとき、処理を行うことが可能か否かを判定し、可能な場合は処理を行うべきサーバを決定し、処理を開始する。もう1台のサーバは、引き継ぎ可能な状態にする。「引き継ぎ可能な状態にする」とは、処理を引き継ぐために必要な情報を、処理をしているサーバから、もう1台のサーバに全て送り、処理をしているサーバに障害が発生しても、もう1台のサーバが処理を続けられるようにすることを指す。

【0041】

（2）2台のサーバに障害が発生し、その後1台のサーバが復帰したとき、復

帰したサーバで処理を行うことが可能か否かを判定し、可能な場合は処理を開始する。

【 0 0 4 2 】

(3) 1 台のサーバが処理中で、もう 1 台のサーバは障害が発生している状態のとき、障害が発生しているサーバが復帰した際に、そのサーバを引継ぎ可能な状態にする。

【 0 0 4 3 】

(4) 1 台のサーバが処理中で、もう 1 台のサーバが引継ぎ可能な状態にあるとき、処理中のサーバで障害が発生した場合は、引き継ぎを行う。

【 0 0 4 4 】

これに加え、次の動作（状態）は必ず考えなければならない。

(5) どのタイミングでも、障害が発生する。

【 0 0 4 5 】

これらの動作（状態）に基づき、図 2 に示す状態遷移図 8 0 0 が作成（用意）される。ここでは、サーバの状態を、マスタ（マスタ状態）M、シングルマスタ（シングルマスタ状態）SM、スレーブ（スレーブ状態）SL、及び停止（停止状態）Xの 4 つの状態に分類している。つまり本実施形態では、従来のマスタMを新たなマスタMとシングルマスタSMという 2 状態に分類し、マスタM、スレーブSL、及び停止Xの 3 つの状態にシングルマスタSMという状態が加えられた点に特徴がある。

【 0 0 4 6 】

サーバの状態は、まず、そのサーバが処理を実行しているか、実行していないかによって分けられ、

処理を行っている：

マスタM、シングルマスタSM

処理を行っていない：

スレーブSL、停止X

のようになる。

【 0 0 4 7 】

更に、「処理を行っている」「処理を行っていない」のそれぞれを、
処理を行っている：

行っている処理を引き継ぐ相手が存在する…マスタ M

行っている処理を引き継ぐ相手が存在しない…シングルマスタ SM

処理を行っていない：

引き継ぎのための情報を受け取っている…スレーブ S L

引き継ぎのための情報を受け取っていない…停止 X

のように分類する。

【 0 0 4 8 】

明らかなように、サーバに障害が発生した場合、処理を行わず、引き継ぎもできないので、当該サーバの状態は停止 X の状態になる。

【 0 0 4 9 】

さて、図 2 の状態遷移図 8 0 0 では、一方のサーバ（例えばサーバ 1 0 0 a）の状態を A、もう一方のサーバ（例えばサーバ 1 0 0 b）の状態を B とすると、（A B）と表記している。（A B）と（B A）は異なる状態を表している。

【 0 0 5 0 】

図 2 の状態遷移図 8 0 0 で適用されるそれぞれの状態について説明する。

まず（X X）は、両サーバが共に停止状態にある状態を示す。具体的には、どちらのサーバも処理も行っておらず、引き継ぎもできない状態を示す。

【 0 0 5 1 】

（SM S L）と（S L SM）とは、一方のサーバが SM（シングルマスタ）状態にあり、他方のサーバが S L（スレーブ）状態にある状態を示す。具体的には、引き継ぎ可能な状態にするために、SM の状態のサーバから、S L の状態のサーバに、処理を引き継ぐための情報を送っている状態を示す。この状態では、一方のサーバで処理は行われているが、他方のサーバでの引き継ぎはできない。

【 0 0 5 2 】

（S L M）と（M S L）とは、一方のサーバが M（マスタ）状態にあり、

他方のサーバが S L (スレーブ) 状態にある状態を示す。具体的には、一方のサーバで処理が行われ、且つ他方のサーバで引き継ぐことができる状態である。

【 0 0 5 3 】

(X S M) と (S M X) とは、一方のサーバが S M (シングルマスタ) 状態にあり、他方のサーバが X (停止) 状態にある状態を示す。具体的には、1 台のサーバが処理を行っているが、もう 1 台のサーバに引き継ぎをするための情報を送ることができない状態である。つまり引き継ぎはできない。

【 0 0 5 4 】

次に、図 2 の状態遷移図 8 0 0 を参照して、上記動作 (1) ~ (5) について説明する。

動作 (1) :

2 台のサーバで共に障害が発生して停止している状態は、(X X) という状態である。ここで、両サーバが障害から復帰すると、図 2 の 1 - 1 - 1 → 1 - 1 - 2、または 1 - 2 - 1 → 1 - 2 - 2 の状態遷移が行われる。

【 0 0 5 5 】

このように、両サーバが障害から復帰した場合には、まず、どちらかのサーバで処理を始める。そして、引き継ぎ可能な状態にするために、引き継ぎに必要な情報を、処理を行っているサーバから、もう 1 台のサーバに送る状態に遷移する (1 - 1 - 1 または 1 - 2 - 1)。これが (S L S M) または (S M S L) 状態である。

【 0 0 5 6 】

その後、引き継ぎが可能になったら、一方はスレーブ状態のまま、もう一方が (シングルマスタ状態から) マスタ状態という状態、つまり (S L M) または (M S L) の状態に移る (1 - 1 - 2 または 1 - 2 - 2)。この状態でも、引き継ぎに必要な情報はマスタからスレーブに送られる。

【 0 0 5 7 】

動作 (2) :

両サーバが障害発生 (= 停止) の状態 (X X) で、一方のサーバだけが障害から復帰した場合は、2 - 1 または 2 - 2 の遷移が行われる。これは、復帰した

サーバを処理のできる状態（シングルマスタの状態）にするための状態遷移であり、（X SM）または（SM X）の状態に移る。

【0058】

動作（3）：

一方のサーバが処理中で、他方のサーバが障害発生（＝停止）という状態は、（X SM）または（SM X）である。この（X SM）または（SM X）の状態、停止状態のサーバが障害から復帰した場合、それぞれ 3-1-1→3-1-2、または 3-2-1→3-2-2 の状態遷移が行われる。

【0059】

このように、（X SM）または（SM X）の状態、停止状態のサーバが障害から復帰した場合には、まず、一方はシングルマスタ状態のまま、もう一方は（停止状態から）スレーブ状態という状態に移り（3-1-1または3-2-1）、引き継ぎに必要な情報を、処理を行っているサーバ（シングルマスタ）から、もう一方のサーバ（スレーブ）に送る。これが（SL SM）または（SM SL）状態である。

【0060】

その後、引き継ぎが可能になったら、動作（1）と同様に、一方はスレーブ状態のまま、もう一方が（シングルマスタ状態から）マスタ状態という状態、つまり（SL M）または（M SL）の状態に移る（3-1-2または3-2-2）。これは、それまで処理を行っていたサーバをマスタ、復帰したサーバをスレーブとして、マスタ、スレーブを確立するための状態遷移である。

【0061】

動作（4）：

引き継ぎが可能な状態ということは、（SL M）または（M SL）という状態にあることである。この状態で、スレーブ状態のサーバに障害が発生した場合は、（X SM）または（SM X）という状態に遷移する（4-1または4-2）。これは、処理はそのまま同じサーバが継続するものの、引き継ぎを行わない状態への状態遷移である。これに対し、マスタ状態のサーバに障害が発生した場合は、（SL M）または（M SL）の状態は（SM X）または（X SM

) の状態に遷移する (4 - 3 または 4 - 4)。これは、それまでスレーブ状態だったサーバが処理を引き継ぐための状態遷移である。

【0 0 6 2】

動作 (5) :

サーバに障害が発生した場合、必ずそのサーバは停止状態 (X) になってしまう。これが 5 - 1 乃至 5 - 8 である。以下にそれぞれの動作を説明する

まず、一方がマスタで、もう一方がスレーブの状態、つまり (S L M) または (M S L) の状態で、両方のサーバに障害が発生した場合には、5 - 1 または 5 - 5 の遷移をして (X X) の状態となる。

【0 0 6 3】

次に、シングルマスタ状態のサーバから引き継ぎを可能にするための情報をスレーブ状態のサーバに送っている (S L S M) または (S M S L) の状態で、シングルマスタ状態のサーバに障害が発生した場合には、引き継ぎはできないため、5 - 2 または 5 - 6 の遷移が行われて、どちらのサーバも停止状態という状態、つまり (X X) の状態となる。また、上記 (S L S M) または (S M S L) の状態で両サーバに障害が発生した場合も、5 - 2 または 5 - 6 の遷移が行われて、両サーバは停止状態になる。

【0 0 6 4】

また、上記 (S L S M) または (S M S L) の状態で、スレーブ状態のサーバに障害が発生した場合は、5 - 3 または 5 - 7 の遷移が行われて (X S M) または (S M X) の状態に移る。この状態では、処理は残ったサーバ (シングルマスタ状態のサーバ) で続行される。

【0 0 6 5】

次に、一方のサーバがシングルマスタ状態で、もう一方のサーバが停止状態、つまり (X S M) または (S M X) の状態で、シングルマスタ状態のサーバに障害が発生した場合には、5 - 4 または 5 - 8 の遷移をして、両サーバ共に停止状態、つまり (X X) となる。

【0 0 6 6】

以上のように、図 2 に示した状態遷移図 8 0 0 には、自動運転を行うための動

作（１）～（５）が全て含まれている。

【 0 0 6 7 】

さて、上記の動作（１）～（５）のうち、動作（３）～（５）については、状態の変わるサーバがどちらであるかが決まっているため、制御することは容易である。

【 0 0 6 8 】

しかし、動作（１）は、処理を引き継ぐサーバが自身であるか否かを判定する必要がある、動作（２）も、障害から復帰したサーバがそのまま処理を引き継いでもよいか否かを判定する必要がある。本実施形態は、この動作（１）及び（２）において処理を引き継ぐサーバを判定する手法を、先に述べたサーバ状態の分類を利用して工夫した点に特徴がある。この判定手法の詳細については別途説明する。ここでは、処理を引き継ぎサーバは、最後まで処理を行っていた方のサーバとなることから、そのことについて述べる。

【 0 0 6 9 】

まず、動作（１）において、処理を引き継ぐべきサーバである条件は、２台のサーバのうち、最後まで処理を行っていた方のサーバであることである。その理由を以下に説明する。

【 0 0 7 0 】

本実施形態において、それまでに処理を行っていなかったサーバが処理の続きを行うためには、処理を行っていたサーバから、引き継ぎのための情報を受け取っておく必要がある。もし、最後に処理を行っていたサーバに障害が発生する直前に、当該サーバからもう一方のサーバに引き継ぎのための情報を送っていなかった場合は、もう一方のサーバでは処理の続きを行うことができない。そのため、最後に処理を行っていた方のサーバを、処理を引き継ぐべきサーバとする必要がある。

【 0 0 7 1 】

次に、最後に処理を行っていたサーバに障害が発生する前に、当該サーバからもう一方のサーバに引き継ぎのための情報を送っていた場合は、どちらのサーバでも処理の続きをすることが可能なように見える。しかし、障害発生直前に行

っていた処理については、当該処理の引き継ぎに必要な情報を上記もう一方のサーバに送る前に、障害が発生している可能性がある。そのため、こちらの場合も、最後に処理を行っていた方のサーバを、処理を引き継ぐべきサーバとする必要がある。

【 0 0 7 2 】

以上から、動作（１）では、最後まで処理を行っていたサーバが処理を引き継ぐ必要がある。

【 0 0 7 3 】

次に動作（２）において、障害から復帰したサーバに処理をさせる条件は、動作（１）の場合と同じ理由により、当該障害復帰サーバが、２台のうち最後まで処理を行っていた方のサーバであることである。

【 0 0 7 4 】

さて、各サーバにおいて、自身が最後まで処理を行っていたサーバであるか否かを判定する手法として、〔従来の技術〕の欄で述べたような「時刻情報を使う方法」がある。しかし、「時刻情報を使う方法」は、時間というグローバルなものを暗黙に仮定し、各サーバの使う時計が常に同期しているという仮定のもとで成立しており、実際の時計は必ずしも常に同期しているとは限らないことから、完全性に問題がある。

【 0 0 7 5 】

このため本実施形態では、最後に処理を行っていたサーバをローカルな情報だけを使って判定する手法を適用している。但し、ローカルな情報には、次のような問題がある。それは、ローカルな情報は、当該情報を持つサーバが停止してしまうと、他のサーバから変更できないため、それぞれのサーバの持つ情報が矛盾してしまう可能性があるという問題である。

【 0 0 7 6 】

この問題について、処理を行っているサーバが、自分で処理を行っている旨の情報をローカルに持つ場合を例に述べる。この例では、処理をしているサーバは「自分が処理中」という情報を持っている。その後、このサーバに障害が発生し、相手サーバが処理を引き継いだ場合、もう一方のサーバも「自分が処理中」と

いう情報を持つことになる。このとき、障害が発生したサーバの持つ「自分が処理中」という情報を書き換えることはできない。このような状態で、今度は、もう一方のサーバにも障害が発生し、その後、両方のサーバが障害から復帰したとする。このとき、両方のサーバが「自分が処理中」という情報を持つことになる。したがって、処理をしているサーバが「自分が処理中」という情報をローカルに持つだけでは、どちらが最後に処理をしていたかを判定できない。

【0077】

そこで本実施形態では、シングルマスタ状態を含む4つの状態に分類される各サーバの状態を、図2に示した状態遷移図800に基づいて制御し、後述するサーバ優先度と称する3つの状態をとり得る状態変数をローカルな情報として持つことにより、この問題を解決するようにしている。

【0078】

ここで、再び図1を参照すると、サーバ100a, 100bは同一構成を有している。即ちサーバ100a, 100bは、いずれもクラスタ管理部110、処理・引き継ぎ部400、及びサーバ優先度処理部700を備えている。これら各部110, 400, 700は、サーバ100a, 100bが所定のソフトウェアプログラムを読み込み実行することにより実現される機能手段である。ここでは、クラスタ管理部110を実現するためのソフトウェア（クラスタソフトウェア）と、処理・引き継ぎ部400を実現するためのソフトウェア（処理ソフトウェア）と、サーバ優先度処理部700を実現するためのサーバ優先度管理ソフトウェアとは、同一の記録媒体、例えばCD-ROMに記録して提供され、サーバ100a, 100bの持つディスク装置200a, 200bにインストールして用いられる。なお、上記各ソフトウェアがディスク装置200a, 200bに予めインストールされていてもよく、また、それぞれ別々の記録媒体に記録されていてもよい。また、ネットワーク500を介してダウンロードされるものであっても構わない。

【0079】

クラスタ管理部110は、図2に示した状態遷移図800に従う状態遷移制御を行う機能を有する。即ちクラスタ管理部110は、もう一方の（サーバ上で動

作する) クラスタ管理部 1 1 0 とネットワーク 5 0 0 を通じて通信することにより、予め定められた状態遷移図 8 0 0 に基づき、2 台のサーバ 1 0 0 a, 1 0 0 b の状態遷移制御を行う。またクラスタ管理部 1 1 0 は、状態遷移の際には、処理・引き継ぎ部 4 0 0 とサーバ優先度処理部 7 0 0 とに、最新の遷移状態を示す状態変化情報 9 0 1 を送る。

【0 0 8 0】

またクラスタ管理部 1 1 0 は、サーバ 1 0 0 a 及び 1 0 0 b の両サーバが停止状態 (X) にある状態で、自身が動作するサーバ (自サーバ) が障害から復帰した際には、サーバ優先度処理部 7 0 0 に自サーバの障害復帰を示す状態変化情報 9 0 1 を送る。更にクラスタ管理部 1 1 0 は、相手サーバのクラスタ管理部 1 1 0 との通信により、相手サーバの障害発生と、障害からの復帰とを検出し、状態遷移を行う。なおクラスタ管理部 1 1 0 は、自サーバの障害復帰時には、復帰後所定時間を待って、相手サーバも障害から復帰したか否かを当該相手サーバとの通信により調べ、しかる後に状態変化情報 9 0 1 を送るようにしている。

【0 0 8 1】

処理・引き継ぎ部 4 0 0 は、クラスタ管理部 1 1 0 から得られる (最新の遷移状態を示す) 状態変化情報 9 0 1 に応じて、処理と処理引き継ぎのための動作の開始/停止の制御を行う。処理・引き継ぎ部 4 0 0 は、処理制御部 4 1 0 と、引き継ぎ制御部 4 2 0 とから構成される。

【0 0 8 2】

処理制御部 4 1 0 は処理の開始/停止を制御し、引き継ぎ制御部 4 2 0 は、処理の引き継ぎのための動作の開始/停止を制御する。処理とは、例えば DBMS (データベースマネジメントシステム) 等のアプリケーションプログラムの実行である。

【0 0 8 3】

サーバ優先度処理部 7 0 0 は、自サーバ 1 0 0 i (i は a または b) のディスク装置 2 0 0 i (に確保された優先度記録領域) に自サーバ 1 0 0 i の優先度 (サーバ優先度) 2 1 0 を記録し、当該優先度 2 1 0 を相手サーバ 1 0 0 j (j は a または b、但し i ≠ j) のディスク装置 2 0 0 j に記録された優先度 2 1 0 と

比較する。このサーバ優先度 2 1 0 は最後まで処理を行っていたサーバを判定するために必要な情報であり、1, 2, 3 の 3 状態のいずれかをとる。

【0 0 8 4】

サーバ優先度処理部 7 0 0 は、状態書き込み部 7 1 0 と、比較処理部 7 2 0 とから構成される。

状態書き込み部 7 1 0 は、クラスタ管理部 1 1 0 から得られる状態変化情報 9 0 1 に応じて、自サーバ 1 0 0 i のディスク装置 2 0 0 i (に確保された優先度記録領域) に記録されている自サーバ 1 0 0 i の優先度 2 1 0 を更新する。

【0 0 8 5】

比較処理部 7 2 0 は、クラスタ管理部 1 1 0 から自サーバの障害復帰を示す状態変化情報 9 0 1 を受け取った場合、相手サーバの比較処理部 7 2 0 との間でネットワーク 5 0 0 を介して優先度 2 1 0 の授受を行う。そして比較処理部 7 2 0 は、自サーバ 1 0 0 i の優先度 2 1 0 と相手サーバ 1 0 0 j の優先度 2 1 0 とを比較することで、自サーバ 1 0 0 i の方が優先度が高いか否かの判定を行う。この判定の詳細は後述する。比較処理部 7 2 0 での優先度比較・判定結果は、クラスタ管理部 1 1 0 で行われる、自サーバ 1 0 0 i が最後に処理をしたサーバであるか否か (つまり自サーバ 1 0 0 i がマスタとなるサーバであるか否か) の判定に用いられる。

【0 0 8 6】

コマンド送信用計算機 3 0 0 は、ユーザから要求された強制開始命令を、ネットワーク 5 0 0 を介して、サーバ 1 0 0 a, 1 0 0 b のうち当該命令の指定するサーバのクラスタ管理部 1 1 0 に送信する。この強制開始命令は、指定のサーバで強制的に処理を開始させる命令である。

【0 0 8 7】

ここで、強制開始命令を用いた処理の強制開始機能を導入する技術的背景について述べる。

サーバ 1 0 0 a 及び 1 0 0 b の両サーバに障害が発生し、その後、一方のサーバ 1 0 0 i だけが障害から復帰した場合、つまり上記動作 (2) の場合、その障害復帰サーバ 1 0 0 i は処理を行うべきサーバではないと判定されることがあり

得る。この場合の動作としては、もう一方の、本来処理を行うべきサーバが復帰するまで待つことが考えられる。

【0088】

しかし、もう一方のサーバが復帰するまで処理が停止したままであるよりは、処理の続きができなくても構わないので、先に障害から復帰したサーバ100iで処理を再開した方が都合がよい場合も当然あり得る。

【0089】

そこで本実施形態では、このような場合に、先に障害復帰したサーバで強制的に処理を開始できる強制開始モードと、本来処理を行うべきサーバが障害から復帰するまで待つ待機モードの2種のモードの中から、いずれか一方のモードが選択指定できるようにしている。ここでは、通常は待機モードに自動設定され、コマンド送信用計算機300から強制開始命令が与えられた場合だけ、強制開始モードに切り替えられる構成を適用している。

【0090】

次に、サーバ100a, 100bの動作の詳細について、サーバ優先度処理部700の動作を中心に説明する。

処理・引き継ぎ部400は、クラスタ管理部110から最新の遷移状態を示す状態変化情報901が送られると、当該状態変化情報901を受け取り、当該状態変化情報901の示す最新の状態に従って、以下に述べる、処理の開始もしくは停止、または引き継ぎのための動作の開始もしくは停止を行う。

【0091】

まず、処理の開始、停止については、前述したマスタ、シングルマスタ、スレーブ、停止の各状態の説明に従い、処理・引き継ぎ部400内の処理制御部410が次のように動作する。

【0092】

自サーバ100iがマスタ(M)、またはシングルマスタ(SM)の状態となったときは処理を行う。

これに対し、自サーバ100iがスレーブ(SL)、または停止状態(X)となったときは処理を行わない。

【0093】

次に、引き継ぎのための動作の開始、停止については、処理・引き継ぎ部400内の引き継ぎ制御部420が次のように動作する。

一方がシングルマスタまたはマスタ、もう一方がスレーブの状態になった状態では、処理引き継ぎのための情報の授受をネットワーク500を介して行う。ここでは、シングルマスタまたはマスタ側の引き継ぎ制御部420が引き継ぎのための情報の送り手となり、スレーブ側の引き継ぎ制御部420が当該情報の受け手となる。

他の状態の組み合わせでは、引き継ぎのための情報授受は行わない。

【0094】

次に、サーバ優先度処理部700の動作の詳細について、サーバ優先度処理部700を構成する状態書き込み部710と比較処理部720のそれぞれの動作に分けて順に説明する。

まず、サーバ優先度処理部700内の状態書き込み部710は、クラスタ管理部110から最新の遷移状態を示す状態変化情報901が送られた場合、当該状態変化情報901の示す最新の状態に従って、自サーバ100iのディスク装置200i（に確保された優先度記録領域）に記録されているサーバ優先度210を変更するサーバ優先度変更（記録）処理を、図3のフローチャートに従って次のように行う。

【0095】

自サーバ100iの状態がシングルマスタ状態（SM）に遷移したときは、1に変更する（ステップS1，S2）。

自サーバ100iの状態がマスタ状態（M）に遷移したときは、2に変更する（ステップS1，S3）。

自サーバ100iの状態がスレーブ状態（SL）に遷移したときは、3に変更する（ステップS1，S4）。

自サーバ100iの状態が停止状態（X）に遷移したときは、変更しない（ステップS1，S5）。

【0096】

ここで、サーバ優先度 210 の初期値は、サーバ 100a と 100b とで異なり、一方のサーバでは 2 に、もう一方のサーバでは 3 に設定される。このサーバ優先度 210 の初期設定は、自動運転を始める前の処理で行われる。

【0097】

次に、サーバ優先度処理部 700 内の比較処理部 720 の動作、つまり最後に処理を行っていたサーバをクラスタ管理部 110 にて判定するのに必要な優先度比較・判定動作について、強制開始を行わない場合と行う場合とに分けて順次説明する。

【0098】

まず、強制開始を行わない場合の動作を、図 4 のフローチャートを参照して説明する。

図 2 に示した状態遷移図 800 において、最後に処理を行っていたサーバを判定する必要があるのは、1-1-1→1-1-2 または 1-2-1→1-2-2 の遷移の場合と、2-1 または 2-2 の遷移の場合の 2 つ、つまり 2 台のサーバ 100a, 100b に障害が発生して共に停止状態 (X X) となっているときに、両サーバが障害から復帰した場合と、いずれか一方のサーバが障害から復帰した場合である。なお、以下の説明では、1-1-1→1-1-2, 1-2-1→1-2-2 を、その共通部分をとって 1-1, 1-2 で表現する。

サーバ 100a, 100b のクラスタ管理部 110 は、(X X) の状態で自サーバが障害復帰すると、その旨を示す状態変化情報 901 を自サーバのサーバ優先度処理部 700 に送る。

【0099】

サーバ優先度処理部 700 内の比較処理部 720 は、自サーバのクラスタ管理部 110 から自サーバが障害復帰したことを示す状態変化情報 901 が送られた場合、上記 1-1 もしくは 1-2 の遷移、または 2-1 もしくは 2-2 の遷移のために、自サーバが最後に処理を行っていたか否かの判定 (自サーバが処理を引き継ぐか否かの判定) に必要なサーバ優先度 210 の比較・判定が要求されたものと判断し、当該比較・判定を次のように行う。

【0100】

1-1もしくは1-2の遷移となる場合：

(1) 比較処理部720は、相手サーバ（サーバ100jとする）の比較処理部720とネットワーク500を介して通信を行ってサーバ優先度210を授受し、自サーバ（サーバ100iとする）と相手サーバ100jとで、どちらのサーバ優先度210が高いかを比較・判定する（ステップS11, S12）。ここでサーバ優先度210は1が最も高く、2, 3の順に低くなる。つまり、サーバ優先度210は1で最高優先度を、2で2番目の優先度を、3で最低優先度を示す。なお、自サーバ100iのサーバ優先度210が1の場合には、相手サーバ100jのサーバ優先度210を取得することなく、自サーバ100iの方が優先度が高いと判定することも可能である。この理由については後述する。

【0101】

(2) 比較処理部720は、サーバ優先度210の比較・判定結果、つまり自サーバ100iの方が優先度が高いか否かのサーバ優先度判定結果を、自サーバのクラスタ管理部110から（X X）状態で送られた状態変化情報901に対する応答として、当該クラスタ管理部110に通知する（ステップS13）。

【0102】

(3) クラスタ管理部110は、比較処理部720からサーバ優先度判定結果を受け取ると、当該判定結果と相手サーバ100jも障害復帰しているか否かと状態遷移図800とに基づき、図5のフローチャートに従って次のように状態遷移する。

【0103】

まずクラスタ管理部110は、比較処理部720からのサーバ優先度判定結果により自サーバ100iの方がサーバ優先度210が高いと判定できる場合（ステップS21, S22）、自サーバ100iが最後に処理を行っていたサーバであると判断する。しかも、相手サーバ100jも障害復帰している場合（ステップS23）、クラスタ管理部110は、最終的に自サーバ100iがマスタとなるために、まずシングルマスタ状態（SM）となり、相手サーバ100jがスレーブ状態（SL）となるように遷移を行う（ステップS24）。これに対し、相手サーバ100jの方がサーバ優先度210が高い場合には（ステップS25）

、自サーバ 1 0 0 i は最後に処理を行っていたサーバではなく、最終的に相手サーバ 1 0 0 j がマスタとなるものとして、自サーバ 1 0 0 i がスレーブ (SL) 状態となり、相手マスタ 1 0 0 j がシングルマスタ状態 (SM) となるように遷移を行う (ステップ S 2 6)。つまり、図 2 の 1 - 1 - 1 (SL SM) または 1 - 2 - 1 (SM SL) の状態遷移が行われる。なお、相手サーバ 1 0 0 j が障害復帰していない場合には (ステップ S 2 3)、後述する 2 - 1 (X SM) もしくは 2 - 2 (SM X) の遷移となる (ステップ S 2 7)。

【0 1 0 4】

(4) クラスタ管理部 1 1 0 は、この状態遷移結果、つまり最新の遷移状態 (SL SM) または (SM SL) を示す状態変化情報 9 0 1 を処理・引き継ぎ部 4 0 0 及びサーバ優先度処理部 7 0 0 に送る。この際の処理・引き継ぎ部 4 0 0 (内の処理制御部 4 1 0 及び引き継ぎ制御部 4 2 0) と、サーバ優先度処理部 7 0 0 内の状態書き込み部 7 1 0 の動作内容は、先の説明から明らかであり、(サーバ 1 0 0 i または 1 0 0 j に再び障害が発生しないならば) 最終的にはサーバ優先度 2 1 0 の高い方のサーバがマスタとなるように遷移する。つまり、図 2 の 1 - 1 - 2 (SL M) または 1 - 2 - 2 (M SL) の状態に遷移する。

【0 1 0 5】

2 - 1 もしくは 2 - 2 の遷移となる場合：

(1) 比較処理部 7 2 0 は、相手サーバ 1 0 0 j の比較処理部 7 2 0 とネットワーク 5 0 0 を介して通信を行ってサーバ優先度 2 1 0 を授受し、自サーバ 1 0 0 i と相手サーバ 1 0 0 j とで、どちらのサーバ優先度 2 1 0 が高いかを比較する。ところが、2 - 1 もしくは 2 - 2 の遷移となる場合には、相手サーバ 1 0 0 j はまだ停止状態 (X) にあることから、相手サーバ 1 0 0 j のサーバ優先度 2 1 0 を取得することができない。しかし、自サーバ 1 0 0 i のサーバ優先度 2 1 0 が 1 であれば、後述するように自サーバ 1 0 0 i の方が優先度が高いと判断できる。

【0 1 0 6】

(2) そこで比較処理部 7 2 0 は、相手サーバ 1 0 0 j が停止状態にあるために当該相手サーバ 1 0 0 j のサーバ優先度 2 1 0 が取得できない場合には (ステ

ップS11)、自サーバ100iのサーバ優先度210が1(つまり最高優先度)であるか否かを判定する(ステップS14)。もし、自サーバ100iのサーバ優先度210が1の場合には、比較処理部720は相手サーバ100jのサーバ優先度210との比較を行うことなく、自サーバ100iの方がサーバ優先度210が高いことを示すサーバ優先度判定結果を、自サーバ100iのクラスタ管理部110に通知する(ステップS15, S13)。

【0107】

これに対し、自サーバ100iのサーバ優先度210が1でない場合には(ステップS14)、自サーバ100iのサーバ優先度210だけでは、自サーバの方が優先度が高いか否かを判定できないとして、判定不可(不定)を示す優先度判定結果を自サーバ100iのクラスタ管理部110に通知する(ステップS16, S13)。なお、自サーバ100iのサーバ優先度210が3の場合には、自サーバ100iの方が優先度が低い、つまり相手サーバ100jの方が優先度が高いと判定できる。しかし、この例のように相手サーバ100jが停止状態にある場合には状態遷移できない。そこで本実施形態では、相手サーバ100jが障害復帰していない場合、自サーバ100iのサーバ優先度210が1でない限り、全て判定不可として扱うようにしている。

【0108】

(3) クラスタ管理部110は、比較処理部720からサーバ優先度判定結果を受け取ると、当該判定結果と相手サーバ100jも障害復帰しているか否かと状態遷移図800とに基づき、図5のフローチャートに従って次のように状態遷移する。

【0109】

まずクラスタ管理部110は、相手サーバ100jが障害復帰していない状態で、自サーバ100iの方がサーバ優先度210が高い場合には(ステップS21~S23)、自サーバ100iが最後に処理を行っていたサーバであるとして、状態遷移図800に従い、自サーバ100iがシングルマスタになるように遷移を行う(ステップS27)。これに対し、判定結果が判定不可を示す場合(相手サーバ100jが障害復帰しておらず、且つ自サーバ100iのサーバ優先度

210が1でない場合)には(ステップS25)、相手サーバ100jが障害復帰するまで状態遷移を行わない。

【0110】

図2に示した状態遷移図800に従い、上記の判定に基づいてサーバ状態が遷移したとすると、サーバ優先度210は図6のように遷移する。この図では、一方のサーバの状態をA、サーバ優先度をa、もう一方のサーバの状態をB、サーバ優先度をbとしたとき、(A B)と表記し、その下に[a b]と表記している。(A B)と(B A)は異なる状態を表す。[a b]と[b a]は異なる場合を表す。

【0111】

以下で、上述の判定法によって、最後に処理を行っていたサーバを正しく判定できる理由について説明する。

命題1 サーバ優先度210の高いサーバが最後に処理を行っていたサーバである。

命題2 あるサーバのサーバ優先度210が1であれば、そのサーバは必ず最後に処理を行っていたサーバである。

【0112】

この2つの命題1, 2を証明する。そのためには、次の3つの補題1~3が成り立てばよい。

【0113】

補題1

仮定:

- ・ある時点で、自サーバ100iのサーバ優先度210に1がセットされている。その後、自サーバ100iのサーバ優先度210は変更されておらず、相手サーバ100jがマスタ状態またはシングルマスタ状態になっていない。
- ・相手サーバ100jのサーバ優先度210が1ではない。

結論:

自サーバ100iのサーバ優先度210が1であるとき、最後に処理を行ったのは自サーバ100iである。

【0 1 1 4】

補題 2

仮定：

・ある時点で、自サーバ 1 0 0 i のサーバ優先度 2 1 0 に 2 がセットされている。その後、自サーバ 1 0 0 i のサーバ優先度 2 1 0 は変更されておらず、相手サーバ 1 0 0 j がマスタ状態またはシングルマスタ状態になっていない。

・相手サーバ 1 0 0 j のサーバ優先度 2 1 0 が 3 である。

結論：

自サーバ 1 0 0 i のサーバ優先度 2 1 0 は 2 であり、相手サーバ 1 0 0 j のサーバ優先度 2 1 0 が 3 であるとき、最後に処理を行ったのは自サーバ 1 0 0 i である。

【0 1 1 5】

補題 3

仮定：

両サーバ 1 0 0 i, 1 0 0 j が停止した状態である。

結論：

サーバ優先度 2 1 0 は、互いに異なる値をとる。

【0 1 1 6】

命題 1 の証明：

まず、補題 1, 2, 3 の結論を使い、命題 1 を証明する。

命題 1 を証明するには、サーバ優先度 2 1 0 の全ての組み合わせにおいて、優先度の高いサーバが最後に処理を行っていたことを証明できればよい。

【0 1 1 7】

補題 3 から、両サーバ 1 0 0 i, 1 0 0 j が停止した状態のとき、サーバ優先度 2 1 0 は互いに異なっているから、(1, 2, 3 の 3 状態を持つ) サーバ優先度 2 1 0 の組み合わせは、(1, 2) (1, 3) (2, 3) の 3 通りである。

【0 1 1 8】

補題 1 から、(1, 2) (1, 3) についてはサーバ優先度 2 1 0 が 1 のサーバが最後に処理を行っている。次に補題 2 から、(2, 3) については、サーバ

優先度 2 1 0 が 2 のサーバが最後に処理を行っている。

よって、全ての場合において、サーバ優先度 2 1 0 の高いサーバが最後に処理を行っているといえる。

【0 1 1 9】

命題 2 の証明：

次に、補題 1，3 の結論を使って命題 2 を証明する。

命題 2 の仮定より、自サーバ 1 0 0 i のサーバ優先度 2 1 0 は 1 である。このとき、補題 3 より、相手サーバのサーバ優先度は 1 ではない。このとき補題 1 より、自サーバ 1 0 0 i は必ず最後に処理を行っていたサーバである。

このように、補題 1，2，3 が成り立てば、命題 1 及び 2 は成り立つ。

【0 1 2 0】

次に、補題 1，2，3 それぞれについて、仮定が成り立てば結論が成り立つことの証明を行い、最後に、補題の仮定が成り立つことを説明する。

【0 1 2 1】

補題 1 の仮定から、結論が成り立つことの証明：

自サーバ 1 0 0 i のサーバ優先度 2 1 0 に 1 がセットされたということは、その時点で、自サーバ 1 0 0 i はシングルマスタ状態であったことを意味する。その後、相手サーバ 1 0 0 j は 1 度もマスタあるいはシングルマスタになっていないということは、相手サーバ 1 0 0 j はその時点から 1 度も処理を行っていないということである。よって、最後に処理を行ったのは、自サーバ 1 0 0 i である。

【0 1 2 2】

自サーバ 1 0 0 i のサーバ優先度 2 1 0 に 1 がセットされた時点より後に、当該優先度 2 1 0 が変更されていないということは、自サーバの現在の優先度 2 1 0 は 1 である。

仮定から、現在の相手サーバ 1 0 0 j のサーバ優先度 2 1 0 は 1 ではない。

よって、自サーバ 1 0 0 i のサーバ優先度 2 1 0 が 1 であり、相手サーバ 1 0 0 j のサーバ優先度 2 1 0 が 1 ではないとき、最後に処理を行ったのは自サーバ 1 0 0 i である。

【0123】

補題2の仮定から、結論が成り立つことの証明：

自サーバ100iのサーバ優先度210に2がセットされたということは、その時点で、自サーバ100iはマスタ状態であったことを意味する。マスタ状態になるためには、相手サーバ100jは必ずスレーブ状態である必要があり、したがって、その時点での相手サーバ100jのサーバ優先度210は3である。

【0124】

その後、相手サーバ100iは1度もマスタあるいはシングルマスタになっていないということは、相手サーバ100jはその時点から1度も処理を行っていないということである。よって、最後に処理を行ったのは、自サーバ100iである。

【0125】

また、相手サーバ100jのサーバ優先度210は、マスタ或いはシングルマスタにならなければ変わらないので、相手サーバ100jのサーバ優先度210は3のままである。

【0126】

また、自サーバ100iのサーバ優先度210に2がセットされた時点より後に当該優先度210が変更されていないということは、自サーバ100iの現在の優先度210は2である。

【0127】

よって、自サーバ100iのサーバ優先度210が2であり、相手サーバ100jのサーバ優先度210が3であるとき、最後に処理を行ったのは自サーバ100iである。

【0128】

補題3の仮定から、結論が成り立つことの証明：

両サーバ100i, 100jが停止した状態のとき、サーバ優先度210は常に互いに異なる状態をとることを証明するには、サーバ優先度210が常に互いに異なる状態をとることを証明すればよい。そのためには、次の4つの仮定3-1～3-4を証明すればよい。

【 0 1 2 9 】

仮定 3 - 1 :

サーバ優先度 2 1 0 の初期値は互いに異なっている。

仮定 3 - 2

片方のサーバのサーバ優先度 2 1 0 が 1 のとき、もう片方のサーバのサーバ優先度 2 1 0 は 1 にならない。

仮定 3 - 3

片方のサーバのサーバ優先度 2 1 0 が 2 のとき、もう片方のサーバのサーバ優先度 2 1 0 は 2 にならない。

仮定 3 - 4

片方のサーバのサーバ優先度 2 1 0 が 3 のとき、もう片方のサーバのサーバ優先度 2 1 0 は 3 にならない。

これらが証明できれば、片方のサーバのサーバ優先度 2 1 0 が何であっても、もう片方のサーバのサーバ優先度 2 1 0 は同じにならないことがいえる。

【 0 1 3 0 】

仮定 3 - 1 は、サーバ優先度 2 1 0 の動作より明らかである。

以下で、仮定 3 - 2 ~ 3 - 4 を証明する。

仮定 3 - 2 の証明 :

片方のサーバ (サーバ 1 0 0 p とする) のサーバ優先度 2 1 0 が 1 になったとき、もう片方のサーバ (サーバ 1 0 0 q とする) は停止状態である。この状態で、サーバ 1 0 0 p のサーバ優先度 2 1 0 を変えずにサーバ 1 0 0 q のサーバ優先度 2 1 0 を 1 にするには、サーバ 1 0 0 q をシングルマスタ状態にしなければならない。そのためには、両サーバが一旦、停止状態に遷移し、更にサーバ 1 0 0 p がスレーブ状態、サーバ 1 0 0 q が停止状態という状態に遷移しなければならない。しかし、この時点で、サーバ 1 0 0 p のサーバ優先度 2 1 0 は 1 であり、これは両サーバが停止した状態から、マスタスレーブ状態への状態遷移の決定方法に矛盾する。よって、自サーバのサーバ優先度 2 1 0 を変更せずに相手サーバがシングルマスタ状態になることは不可能である。つまり、片方のサーバ 1 0 0 p のサーバ優先度 2 1 0 が 1 のとき、もう片方のサーバ 1 0 0 q のサーバ優先

度 2 1 0 は 1 にならない。

【 0 1 3 1 】

仮定 3 - 3 の証明：

片方のサーバ 1 0 0 p のサーバ優先度 2 1 0 が 2 になったとき、もう片方のサーバ 1 0 0 q は必ず 3 になる。よって、片方のサーバ 1 0 0 p のサーバ優先度が 2 のとき、もう片方のサーバ 1 0 0 q のサーバ優先度は 2 にならない。

【 0 1 3 2 】

仮定 3 - 4 の証明：

片方のサーバ 1 0 0 p のサーバ優先度 2 1 0 が 3 になったとき、もう片方のサーバ 1 0 0 q はマスタ状態である。この状態で、サーバ 1 0 0 p のサーバ優先度 2 1 0 を変えずにサーバ 1 0 0 q のサーバ優先度 2 1 0 を 3 にするには、サーバ 1 0 0 q をスレーブ状態にしなければならない。そのためには、両サーバ 1 0 0 p, 1 0 0 q が一旦停止状態に遷移し、さらにサーバ 1 0 0 p が停止状態、サーバ 1 0 0 q がスレーブ状態という状態に遷移しなければならない。しかし、このときの両者のサーバ優先度 2 1 0 は 3 と 2 であり、サーバ 1 0 0 q の方が優先度 2 1 0 が高い。これは両サーバ 1 0 0, 1 0 0 j が停止した状態から、マスタースレーブ状態への状態遷移の決定方法に矛盾する。よって、自サーバのサーバ優先度 2 1 0 を変更せずに相手サーバがスレーブ状態になることは不可能である。つまり、片方のサーバ 1 0 0 p のサーバ優先度 2 1 0 が 3 のとき、もう片方のサーバ 1 0 0 q のサーバ優先度 2 1 0 は 3 にならない。

このように補題 1, 2, 3 について、仮定から、結論が成り立つことが証明できた。

【 0 1 3 3 】

次に補題 1, 2 の仮定が成り立つことを説明する。

補題 1 の仮定が成り立つことの証明：

自サーバ 1 0 0 i のサーバ優先度 2 1 0 に 1 がセットされたとき、相手サーバ 1 0 0 j のサーバ優先度 2 1 0 が 1 でなければ、自サーバ 1 0 0 i のサーバ優先度 2 1 0 を変えずに、相手サーバ 1 0 0 j がマスタ状態またはシングルマスタ状態になることができないことを、図 2 の状態遷移図 8 0 0 を参照して説明する。

【0134】

自サーバ100iのサーバ優先度210が1にセットされた後に、相手サーバ100jがマスタ状態になるためには、自サーバ100iはスレーブ状態になる必要があり、そのとき自サーバ100iのサーバ優先度210は3に変更されてしまう。つまり、自サーバ100iのサーバ優先度210を変更せずに相手サーバ100jがマスタ状態になることは不可能である。よって、自サーバ100iのサーバ優先度210が1にセットされた後、相手サーバ100jはマスタ状態になっていない。

【0135】

自サーバ100iのサーバ優先度210が1にセットされたより後に、相手サーバ100jがシングルマスタになるためには、図2の状態遷移図800から、一旦、相手サーバ100jがマスタ、自サーバ100iがスレーブ、もしくはその逆に、相手サーバ100jがスレーブ、自サーバ100iがマスタという状態になるか、或いは、一旦両サーバ100i, 100j共に停止状態になって、その後、相手サーバ100jがシングルマスタにならないといけない。

【0136】

ところが、一方がマスタで、他方がスレーブになる場合は、自サーバ100iのサーバ優先度210が変わってしまうので不可能である。また、両サーバ共に停止状態になり、相手サーバ100jがシングルマスタになるためには、判定方法の定義から、相手サーバ100jのサーバ優先度210が1でなければならない。これは仮定に反する。よって、自サーバ100iのサーバ優先度210に1がセットされた後、相手サーバ100jはシングルマスタ状態になっていないといえる。

以上のことから、自サーバ100iのサーバ優先度210を変えずに、相手サーバ100jがマスタ状態またはシングルマスタ状態になることはできない。

【0137】

補題2の仮定が成り立つことの証明：

自サーバ100iのサーバ優先度210が2にセットされたとき、自サーバ100iのサーバ優先度210を変えずに、相手サーバ100jがマスタ状態また

はシングルマスタ状態になることができないことを、図2の状態遷移図800を参照して説明する。

【0138】

自サーバ100iのサーバ優先度210が2にセットされた時点での、自サーバ100iの状態は必ずマスタ状態であり、相手サーバ100jの状態は必ずスレーブ状態である。

【0139】

自サーバ100iのサーバ優先度210が2にセットされた後に、相手サーバ100jがマスタ状態になるためには、自サーバ100iはスレーブ状態になる必要があり、そのとき自サーバ100iのサーバ優先度210は3に変更されてしまう。つまり、自サーバ100iのサーバ優先度210を変更せずに相手サーバ100jがマスタ状態になることは不可能である。よって、自サーバ100iのサーバ優先度210が2にセットされた後、相手サーバ100jはマスタ状態になっていない。

【0140】

自サーバ100iのサーバ優先度210に2がセットされた後に、自サーバ100iのサーバ優先度210を変えることなく相手サーバ100jがシングルマスタ状態になることは可能である。しかし、その後、自サーバ100iのサーバ優先度210を変えることなく、相手サーバ100jのサーバ優先度210を3にするためには、図2の状態遷移図800から、両サーバ100i, 100jが一旦停止状態に遷移し、更に自サーバ100iが停止状態、相手サーバ100jがスレーブ状態という状態に遷移しなければならない。ところが、この時点での両者のサーバ優先度210は自サーバ100iが2で相手サーバ100jが1であり、これは両サーバ100i, 100jが停止した状態から、マスタースレーブ状態への状態遷移の決定方法に矛盾する。よって、自サーバ100iのサーバ優先度210を変更せずに、相手サーバ100jがシングルマスタ状態になることは不可能である。つまり、自サーバ100iのサーバ優先度210が2にセットされた後、相手サーバ100jはシングルマスタ状態になっていない。

【0141】

以上により、自サーバ 1 0 0 i のサーバ優先度 2 1 0 が 2 にセットされたより後に、相手サーバ 1 0 0 j はマスタまたはシングルマスタになっていない。

よって、両サーバ 1 0 0 i, 1 0 0 j が停止した状態のとき、どちらかのサーバのサーバ優先度 2 1 0 が 2 で、もう一方のサーバ優先度 2 1 0 が 3 であった場合、サーバ優先度 2 1 0 が 2 のサーバ、つまり優先度が高い方のサーバが最後に処理をしていたサーバであるといえる。

【0 1 4 2】

次に、強制開始を行う場合の動作を説明する。

図 7 に、強制開始がある場合の状態遷移及びサーバ優先度 2 1 0 の遷移を示す。

図 7 に示す強制開始がある場合の遷移が、図 6 に示した強制開始がない場合と相違する点は、(X X) の状態からの 2 - 1 または 2 - 2 の遷移（図 7 において太線の矢印で示す遷移）で、サーバ優先度 2 1 0 の低い方のサーバを、強制的にシングルマスタにする遷移があるということである。

【0 1 4 3】

そのため、(X X), (X SM), (SM X) の 3 状態のとき、[1 1] というサーバ優先度 2 1 0 を持つ場合がある（図において※印が付されている優先度）。

【0 1 4 4】

例えば、(X X) の状態で、サーバ優先度が [1 3] や、[1 2] のとき、(X SM) へ強制開始によって状態遷移を行うことができる。そのとき、サーバ優先度は [1 1] になる。この状態でシングルマスタ状態のサーバに障害が発生した場合、(X X) で [1 1] になる。

【0 1 4 5】

このように、強制開始の結果、両サーバのサーバ優先度 2 1 0 が等しい状態が発生した場合、優先度の高いサーバを判定することができなくなる。したがって、両サーバが障害から復帰した場合に、最後に処理を行っていたサーバを判定することができなくなる。この場合は、もう 1 度強制開始を行う必要がある。

【0 1 4 6】

また、強制開始の結果、上記の如く (X X) のときに [1 1] という状態があるため、一方のサーバが障害から復帰しても、2-1 または 2-2 の遷移を自動的に行うことができなくなる。2-1 または 2-2 の遷移をさせるためには、必ずもう 1 度強制開始を行う必要がある。

【0 1 4 7】

以上に述べたように、本実施形態においては、サーバの状態を、処理を行い、且つ行っている処理を引き継ぐ相手が存在する「マスタ」と、処理は行いが、行っている処理を引き継ぐ相手が存在しない「シングルマスタ」と、処理を行っていないが、引き継ぎのための情報を受け取っている「スレーブ」と、処理を行っておらず、且つ引き継ぎのための情報を受け取っていない「停止」の 4 つの状態に分類して作成された状態遷移図 8 0 0 に従って、サーバ 1 0 0 a, 1 0 0 b の状態遷移を行うと共に、この状態の遷移で決まるサーバ優先度 2 1 0 という状態変数をローカルなディスク装置 2 0 0 a, 2 0 0 b に記録し、当該状態遷移と状態変数（サーバ優先度 2 1 0）を用いて、最後まで処理を行っていたサーバを判定するようにしたので、次のような効果を得ることができる。

まず、強制開始を使わない場合、サーバ 1 0 0 a, 1 0 0 b の片方または両方（つまり少なくとも一方のサーバ）に、どのようなタイミングで障害発生、或いは障害復帰が起こっても、サーバ 1 0 0 a, 1 0 0 b に障害が発生し、復帰したときに、当該サーバ自身が処理を行うべきサーバであるか否かを判定することができる。

【0 1 4 8】

特に本実施形態においては、前記した自動運転を行うための動作（1）～（5）を、2 台のサーバ 1 0 0 a, 1 0 0 b 内で状態遷移のできるクラスタ管理部 1 1 0 を使うことで実現し、動作（1）（2）で必要なサーバの判定を、状態遷移の度にサーバ優先度処理部 7 0 0 内の状態書き込み部 7 1 0 がローカルなディスク装置に遷移状態に対応したサーバ優先度 2 1 0 を記録して、両サーバ 1 0 0 a, 1 0 0 b のサーバ優先度 2 1 0 をサーバ優先度処理部 7 0 0 内の比較処理部 7 2 0 で比較することによって実現しているため、共有ディスク装置や時刻情報を使うことなく、自動運転を実現できる。

【0 1 4 9】

また、強制開始を使う場合、動作（２）において、復帰したサーバが処理を引き継ぐサーバであると判定できない場合でも、必要に応じて強制的に処理を引き継ぐことができる。この場合においても、両サーバが（X X）で〔1 1〕という特別な状態に遷移した場合を除き、自動的に処理を継続することができる。また、上記特別な状態の場合も、再度強制開始を用いて処理を継続することができる。

【0 1 5 0】

【発明の効果】

以上詳述したように本発明によれば、所定の状態遷移図に従う状態の遷移で決まるサーバ優先度という状態変数をローカルな記憶装置に記録し、当該状態遷移と状態変数を用いて、最後まで処理を行っていたサーバを判定するようにしたので、処理引き継ぎの回数の制限をなくして自動運転を実現すると共に、判定精度を向上することができる。

【図面の簡単な説明】

【図 1】

本発明の一実施形態に係る高可用性計算機システムの構成を示すブロック図。

【図 2】

同実施形態で適用される状態遷移図 8 0 0 を示す図。

【図 3】

図 1 中の状態書き込み部 7 1 0 によるサーバ優先順位変更処理を説明するためのフローチャート。

【図 4】

図 1 中の比較処理部 7 2 0 によるサーバ優先度判定処理を説明するためのフローチャート。

【図 5】

図 1 中のクラスタ管理部 1 1 0 による障害復帰時状態遷移処理を説明するためのフローチャート。

【図 6】

各サーバがローカルに持つサーバ優先度を、強制開始がない場合について、図 2 の状態遷移図 8 0 0 と対応付けて示す図。

【図 7】

各サーバがローカルに持つサーバ優先度を、強制開始がある場合について、図 2 の状態遷移図 8 0 0 と対応付けて示す図。

【符号の説明】

1 0 0 a, 1 0 0 b…サーバ

1 1 0…クラスタ管理部（状態遷移手段、最終処理計算機判定手段）

2 0 0 a, 2 0 0 b…ディスク装置（記憶装置）

2 1 0…サーバ優先度

3 0 0…コマンド送信用計算機

4 0 0…処理・引き継ぎ部

4 1 0…処理制御部

4 2 0…引き継ぎ制御部

5 0 0…ネットワーク

7 0 0…サーバ優先度処理部

7 1 0…状態書き込み部

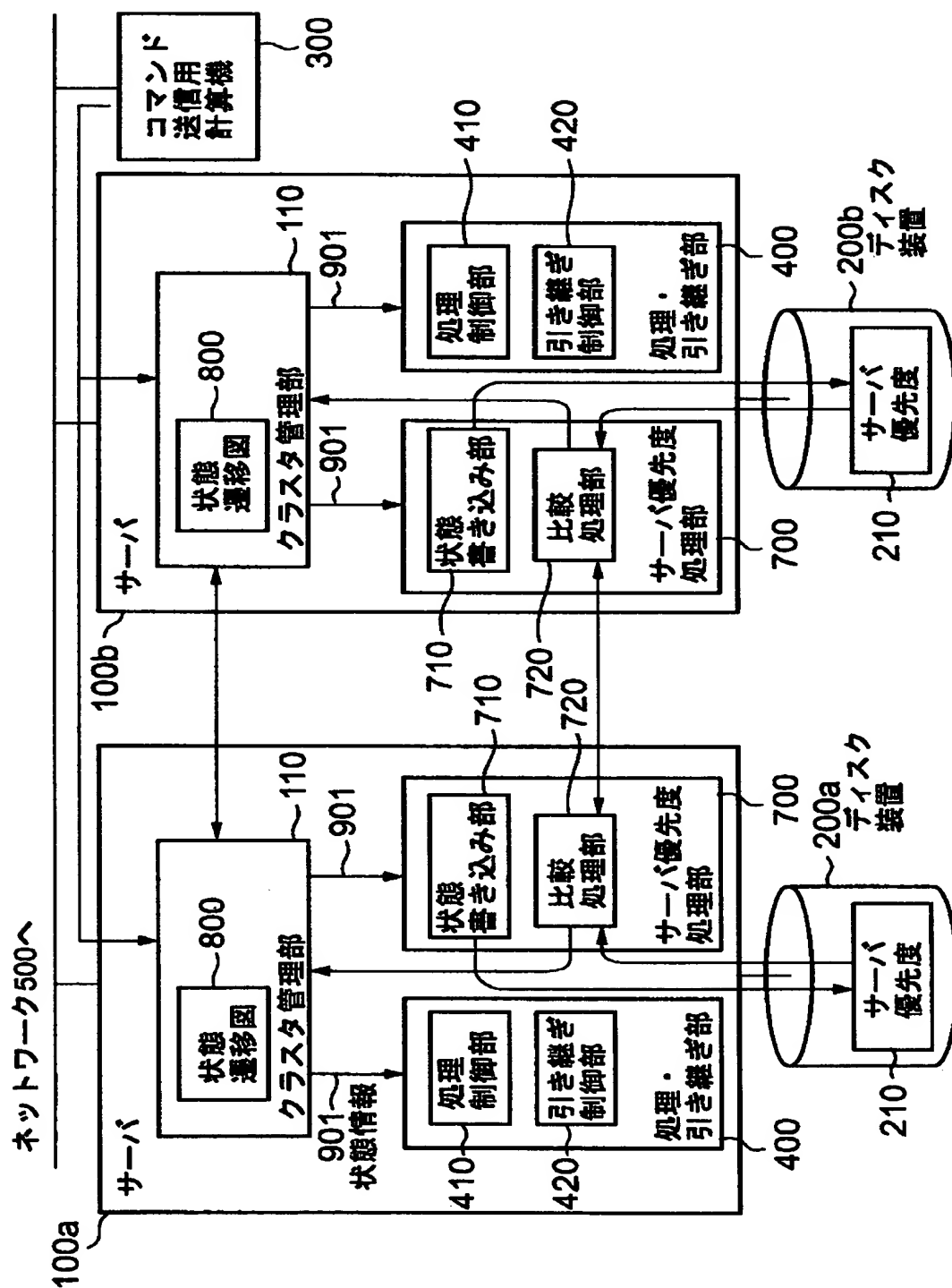
7 2 0…比較処理部（優先度判定手段）

9 0 1…状態変化情報

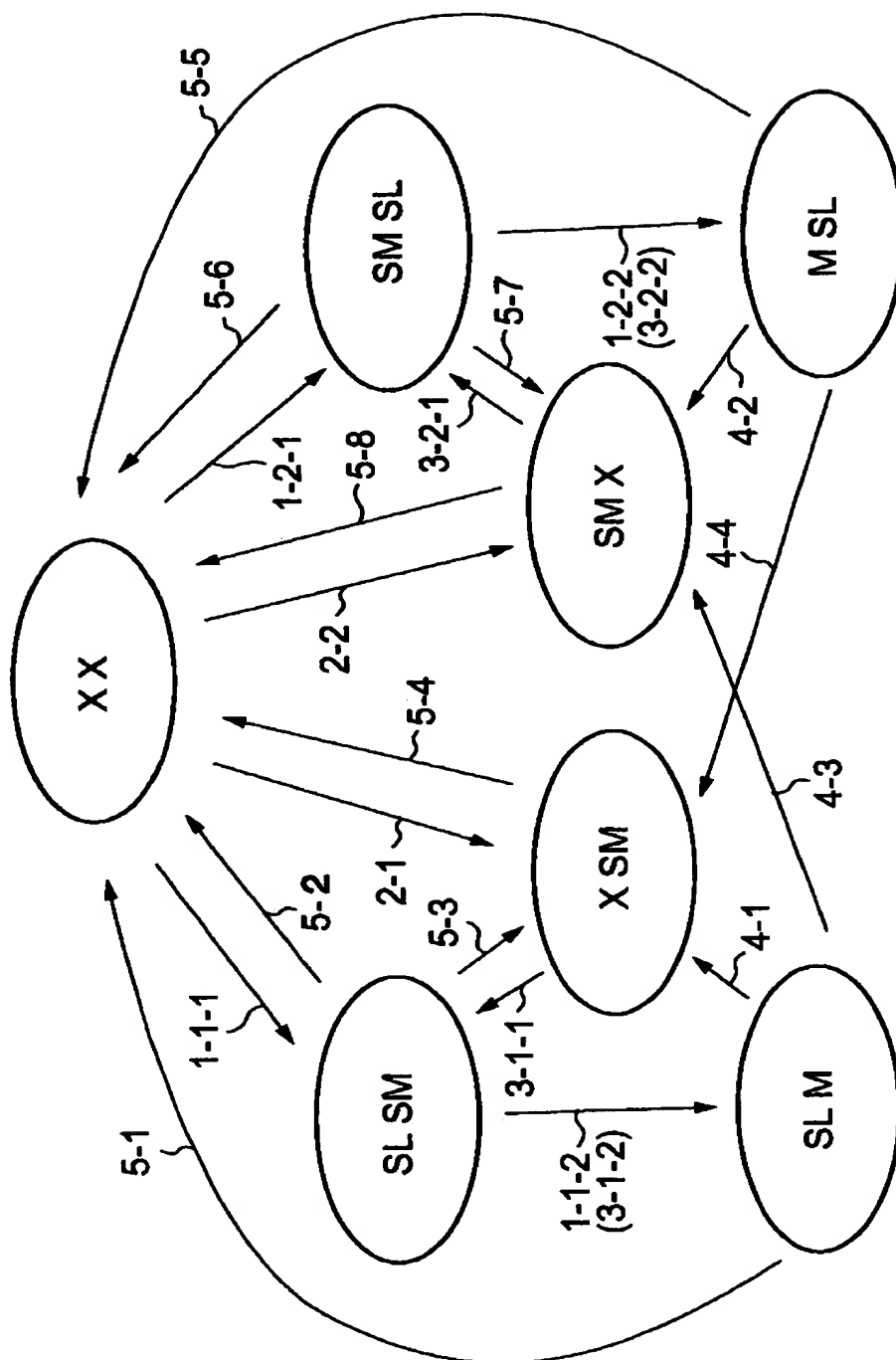
【書類名】

図面

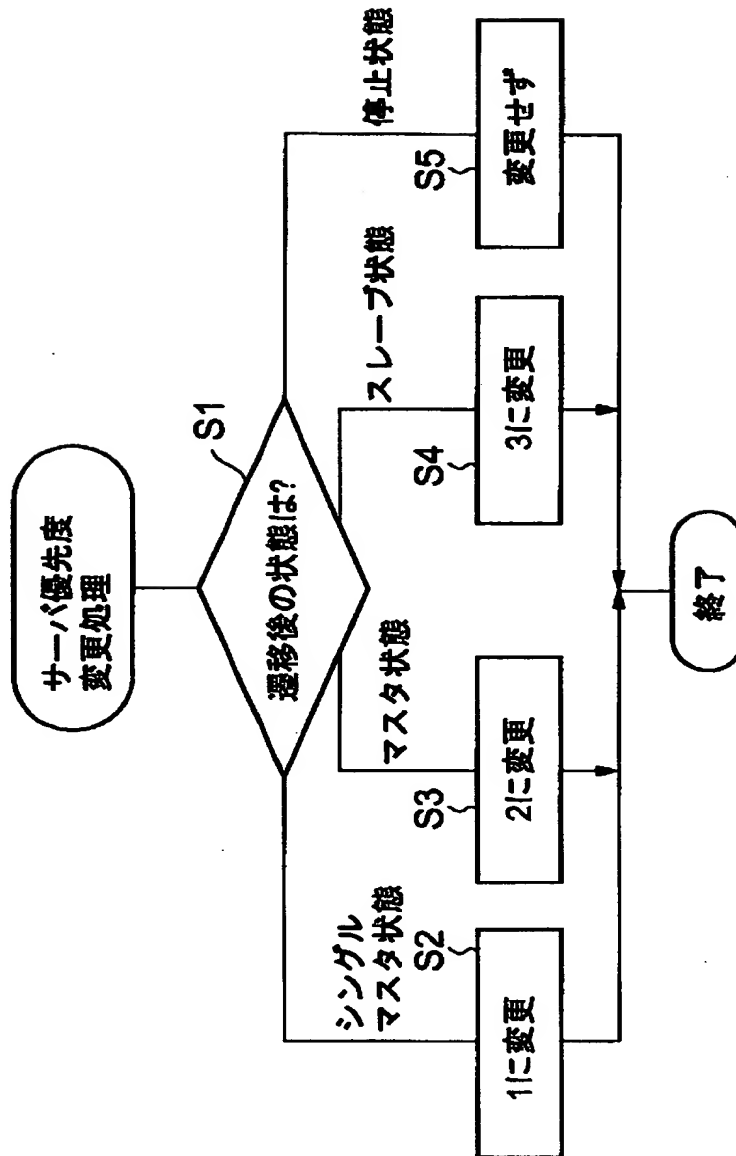
【図 1】



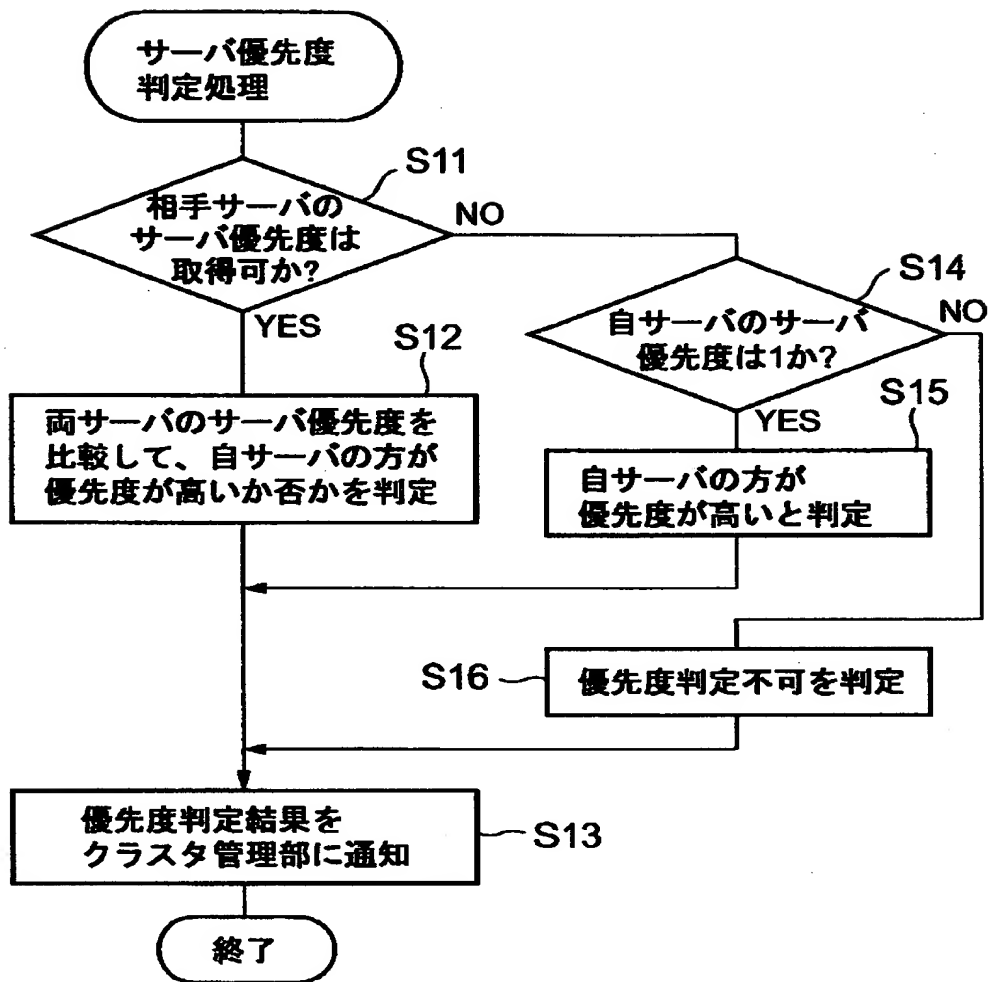
【図 2】



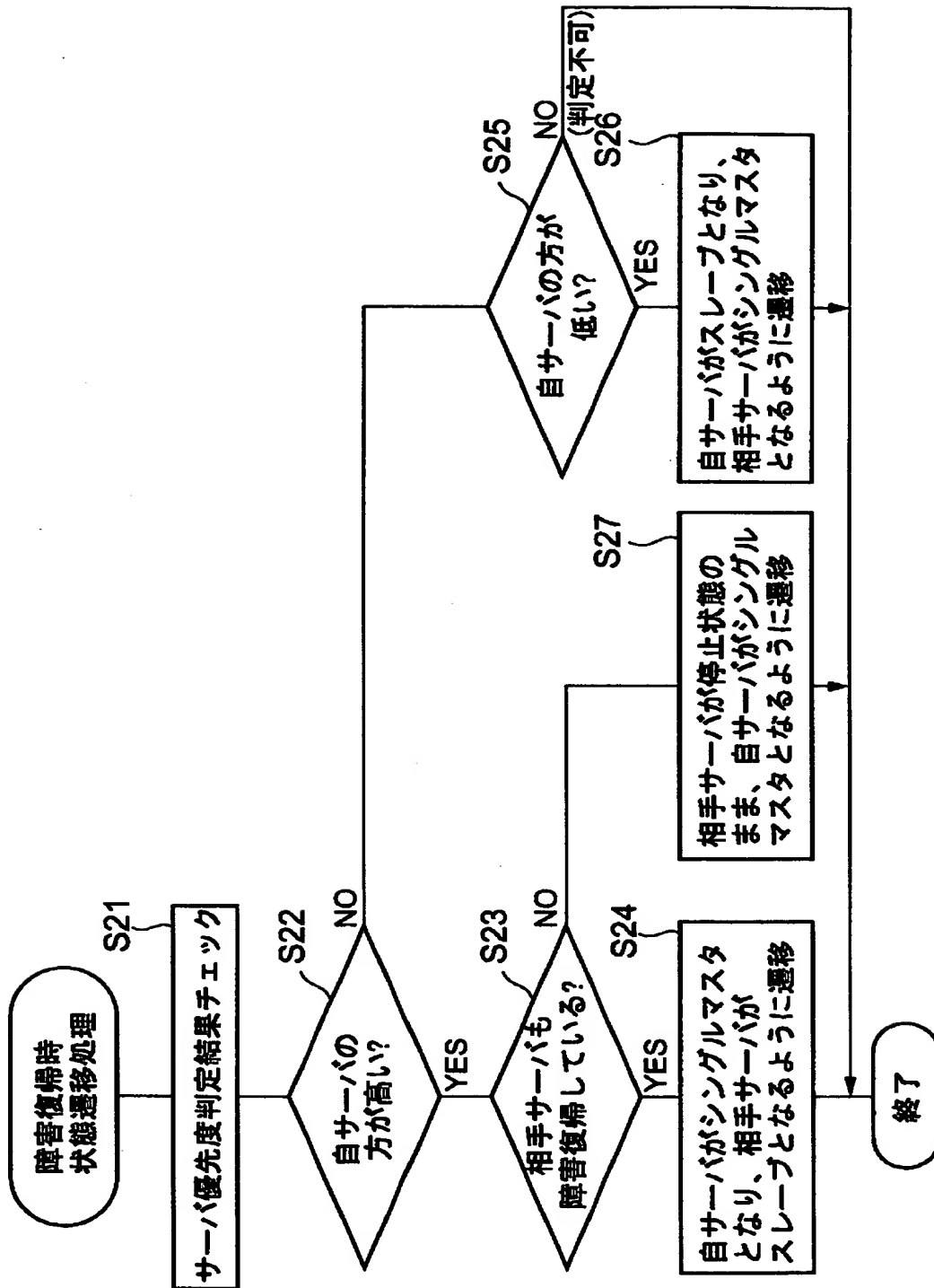
【図 3】



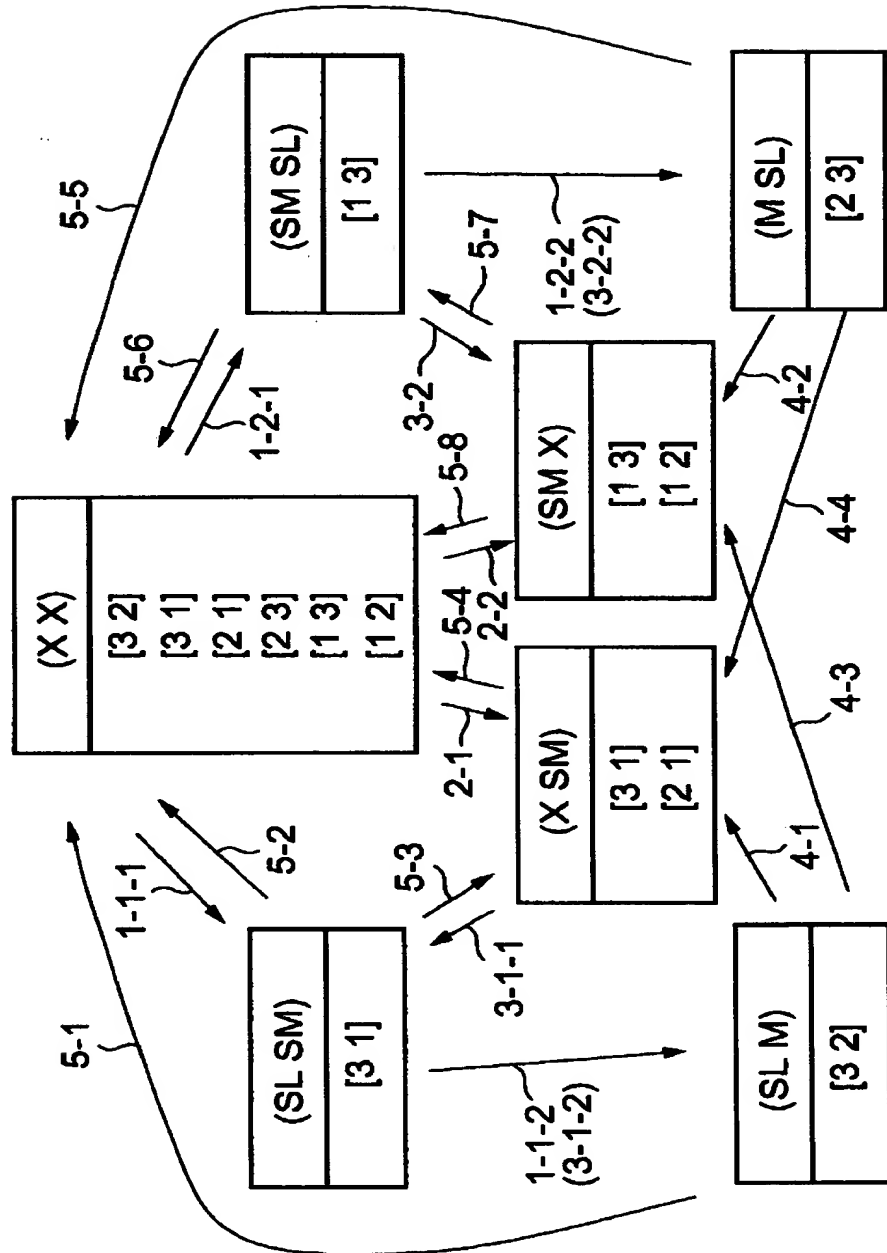
【図 4】



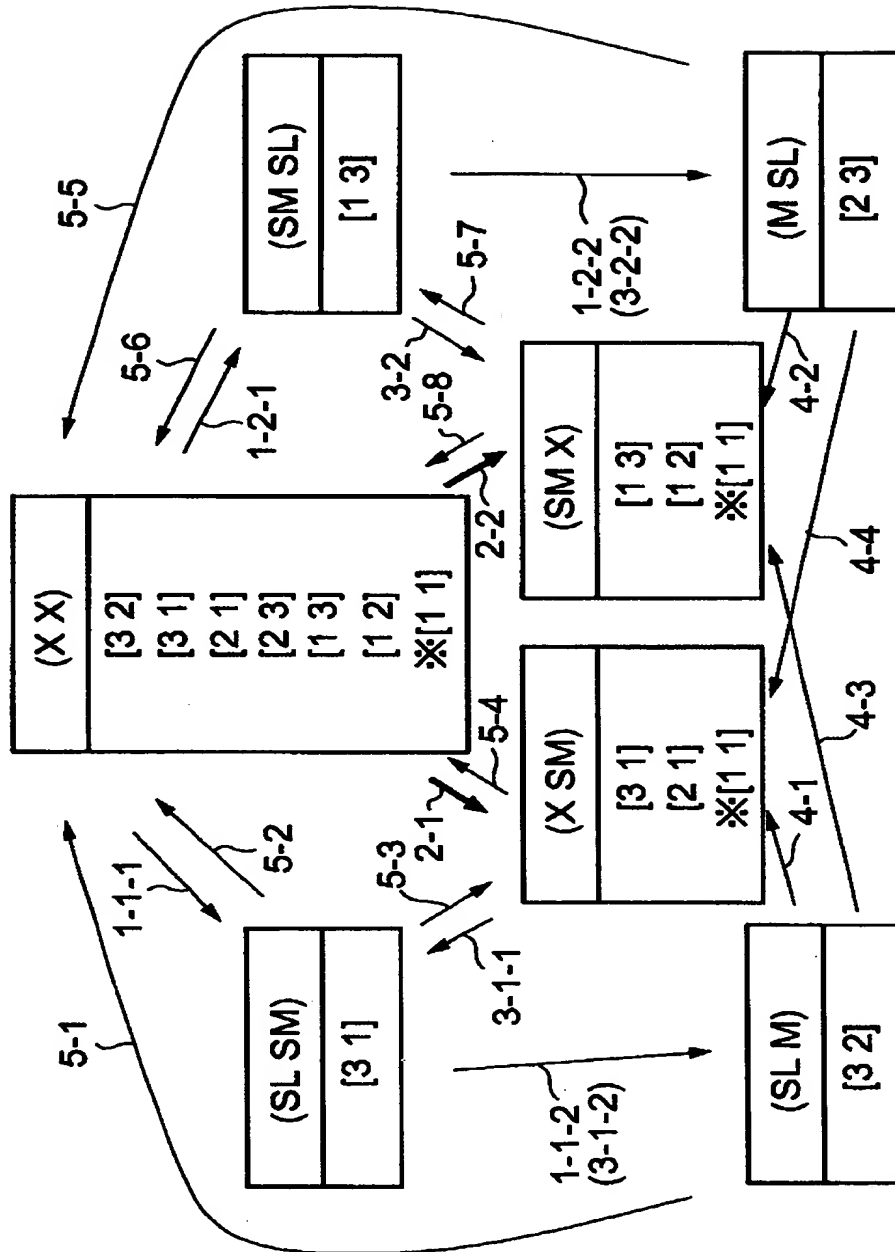
【図 5】



【図 6】



【図 7】



【書類名】 要約書

【要約】

【課題】 ローカルな情報だけを使って最後に処理を行っていたサーバを判定することにより、処理引き継ぎの回数の制限をなくして自動運転を実現すると共に、判定精度の向上を図れるようにする。

【解決手段】 サーバ 1 0 0 a, 1 0 0 b のクラスタ管理部 1 1 0 は、少なくとも一方のサーバで障害が発生した場合と障害から復帰した場合、状態遷移図 8 0 0 に従うサーバ状態の遷移を行う。状態書き込み部 7 1 0 は、状態変化情報 9 0 1 の示すサーバ状態で決まるサーバ優先度 2 1 0 をディスク装置 2 0 0 a, 2 0 0 b に記録する。比較処理部 7 2 0 は、サーバ 1 0 0 a, 1 0 0 b に障害が発生し、その後少なくとも自サーバが障害復帰した際、少なくとも自サーバの優先度 2 1 0 に基づき自サーバの方が優先度が高いかを判定する。クラスタ管理部 1 1 0 は、この判定結果と状態遷移図 8 0 0 に基づき、対応する状態遷移を行う。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [0 0 0 0 0 3 0 7 8]

1. 変更年月日	1 9 9 0 年 8 月 2 2 日
[変更理由]	新規登録
住 所	神奈川県川崎市幸区堀川町 7 2 番地
氏 名	株式会社東芝

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

☐ **BLACK BORDERS**

☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**

☐ **FADED TEXT OR DRAWING**

☒ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**

☐ **SKEWED/SLANTED IMAGES**

☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**

☐ **GRAY SCALE DOCUMENTS**

☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**

☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**

☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.